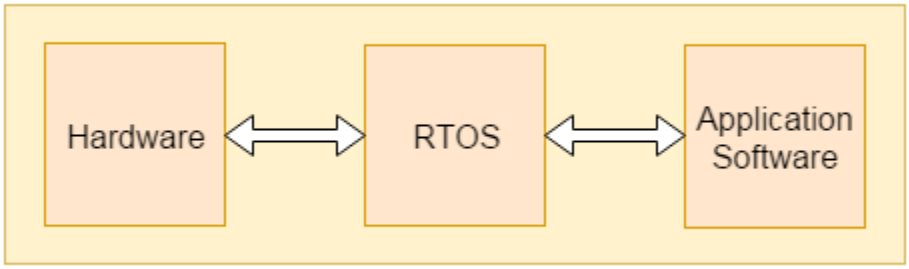
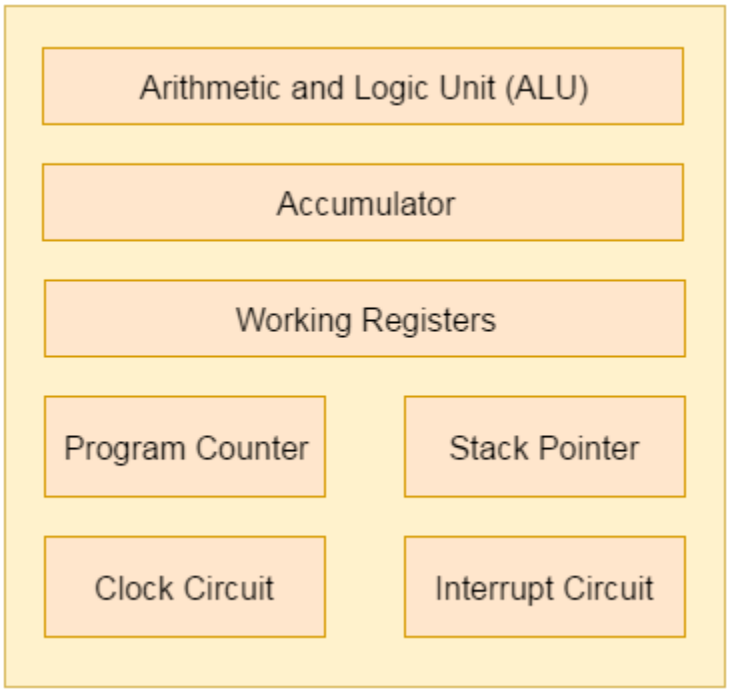
EIOT DIAGRAMS

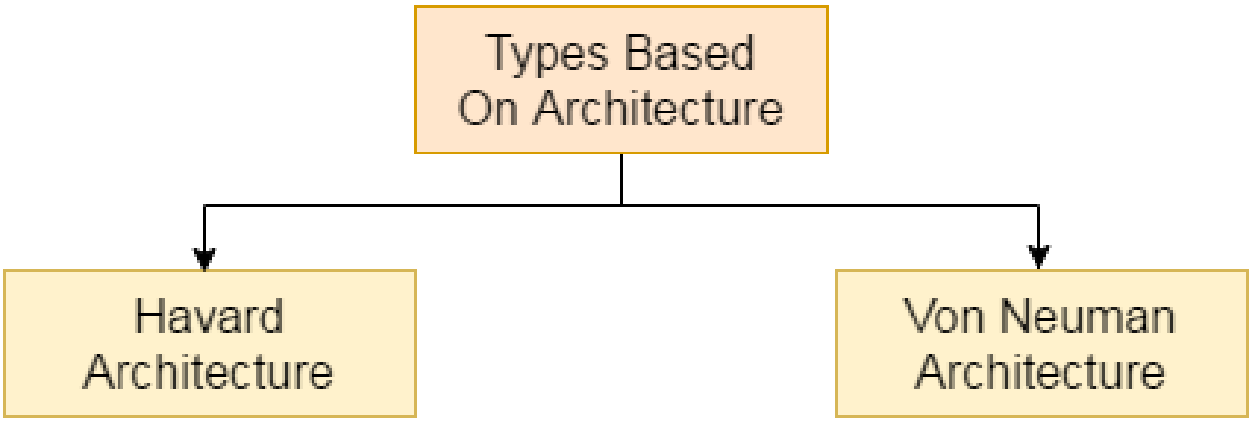
Module 1

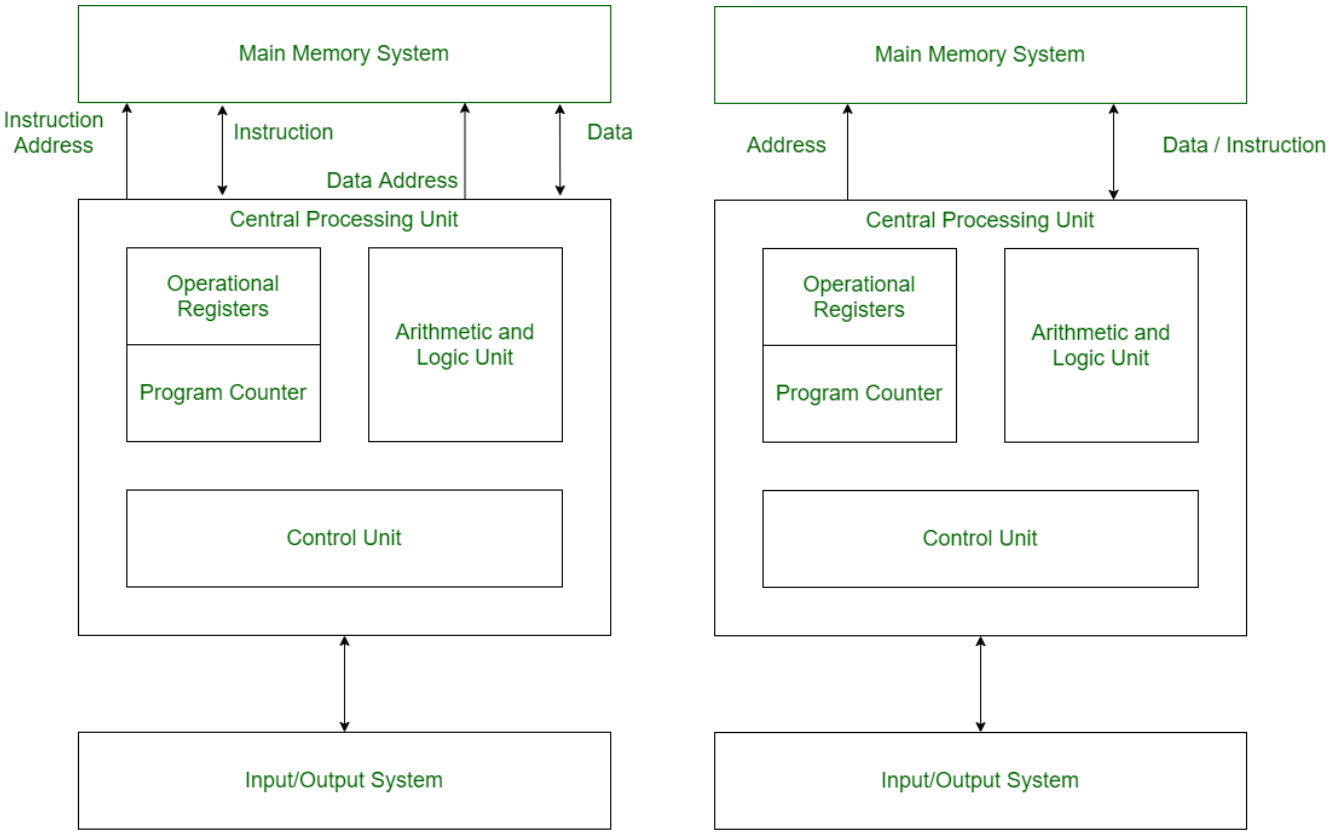


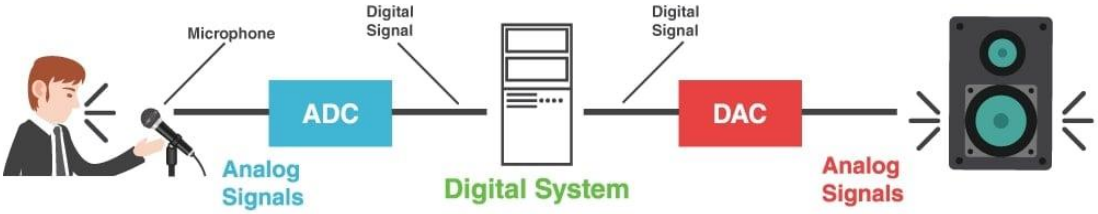
Microproccessor

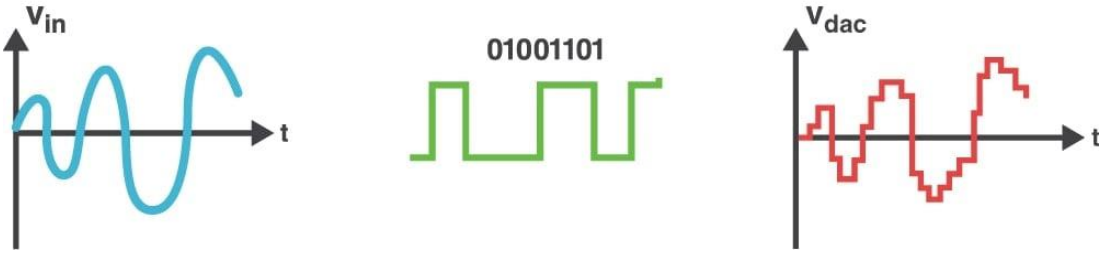


- **Types of Microcontroller**


Types Based
On Architecture
Havard
Architecture
Von Neuman
Architecture


Main Memory System
Instruction Address
Instruction
Data Address
Data
Central Processing Unit
Operational Registers
Program Counter
Arithmetic and Logic Unit
Control Unit
Input/Output System
Harvard Architecture
Main Memory System
Address
Data / Instruction
Central Processing Unit
Operational Registers
Program Counter
Arithmetic and Logic Unit
Control Unit
Input/Output System
Von Neumann Architecture


Microphone
Digital Signal
Digital Signal
ADC
DAC
Analog Signals
Digital System
Analog Signals


Vin
t
01001101
Vdac
t

Register banks in the 8051

Bank 0		Bank 1		Bank 2		Bank 3	
7	R7	F	R7	17	R7	1F	R7
6	R6	E	R6	16	R6	1E	R6
5	R5	D	R5	15	R5	1D	R5
4	R4	C	R4	14	R4	1C	R4
3	R3	B	R3	13	R3	1B	R3
2	R2	A	R2	12	R2	1A	R2
1	R1	9	R1	11	R1	19	R1
0	R0	8	R0	10	R0	18	R0

B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
CY	AC	F0	RS1	RS0	OV	-	P

CY Bit 7 - Carry flag

AC Bit 6 - Auxiliary carry flag for BCD operations

F0 Bit 5 - User defined flag (Flag zero)

RS1, RS0 Bit 4-3 - Select the working register banks as follows :

RS1	RS0	Bank Selection	
0	0	00H - 07H	Bank 0
0	1	08H - 0FH	Bank 1
1	0	10H - 17H	Bank 2
1	1	18H - 1FH	Bank 3

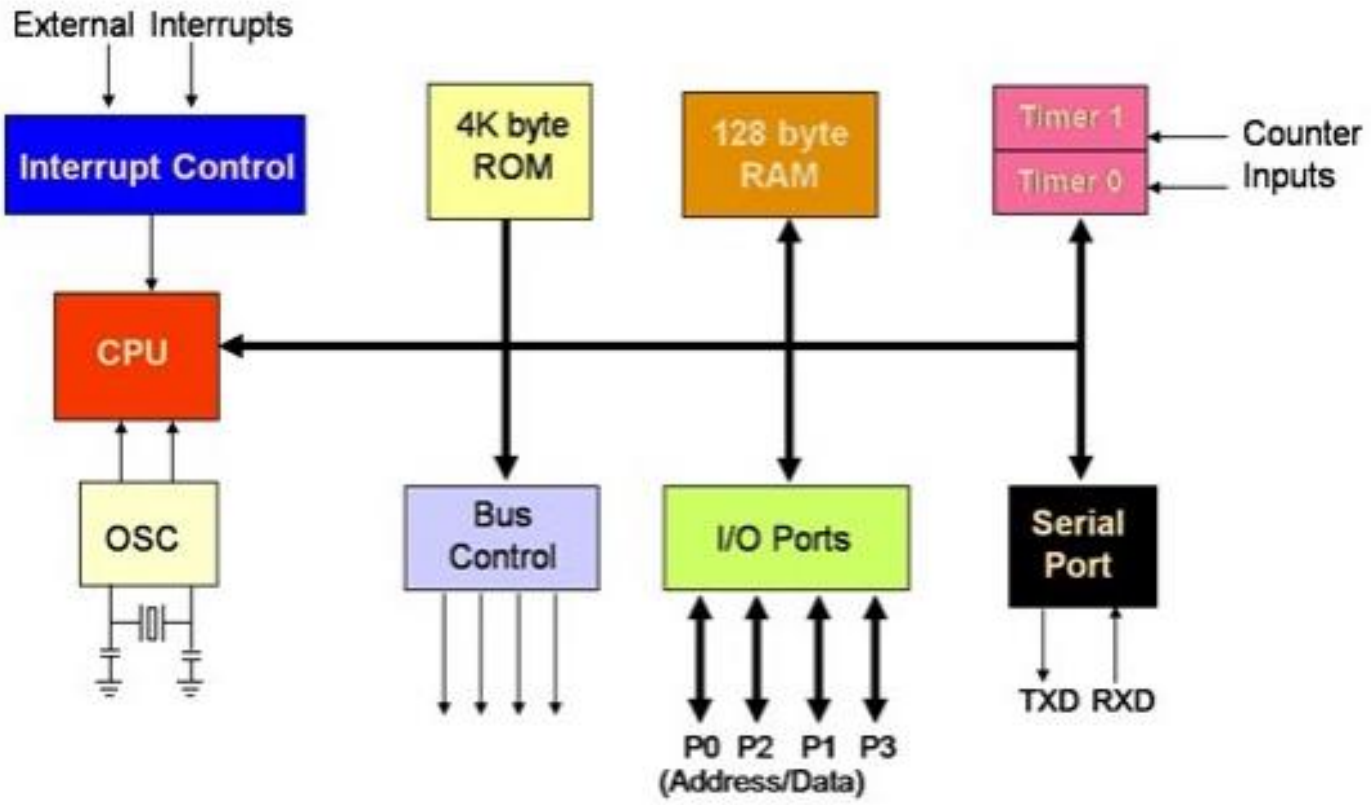
Fig. 12.3 Program status word

OV Bit 2 - Overflow flag

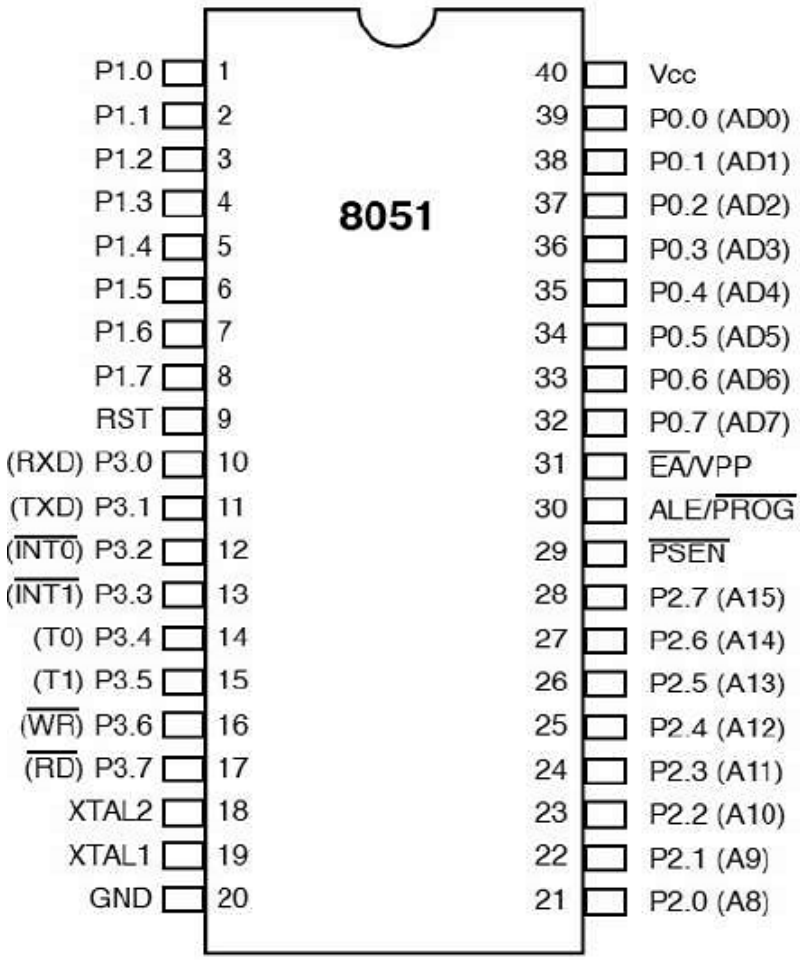
– Bit 1 - Reserved

P Bit 0 - Parity flag (1 = Even parity)

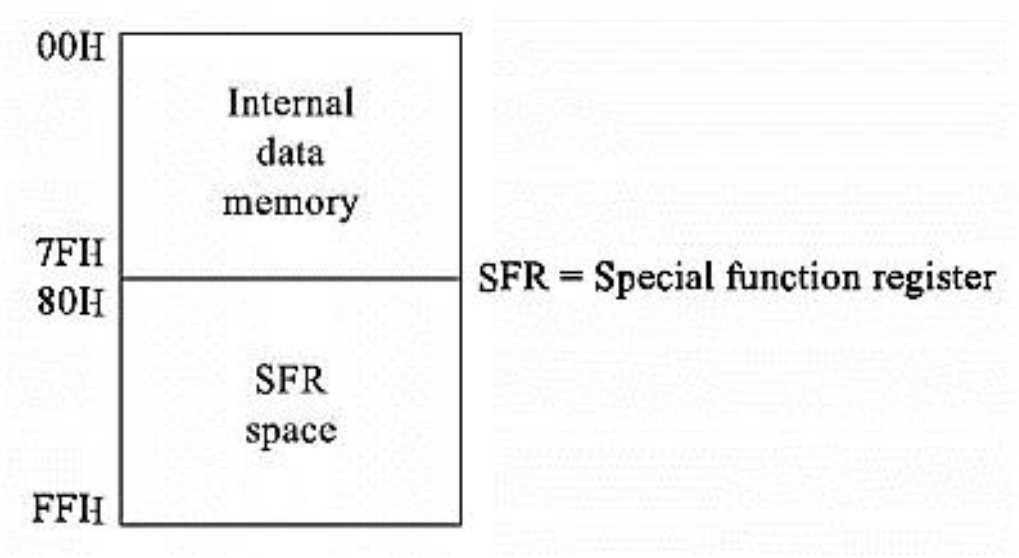
8051 Microcontroller Architecture/Block Diagram



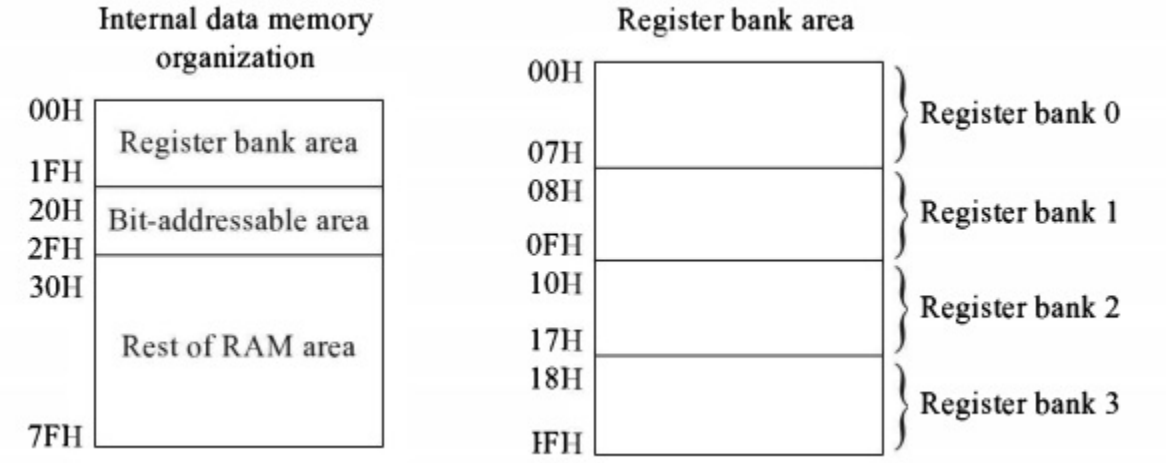
pin diagram of 8051 microcontroller consists of 40 pins



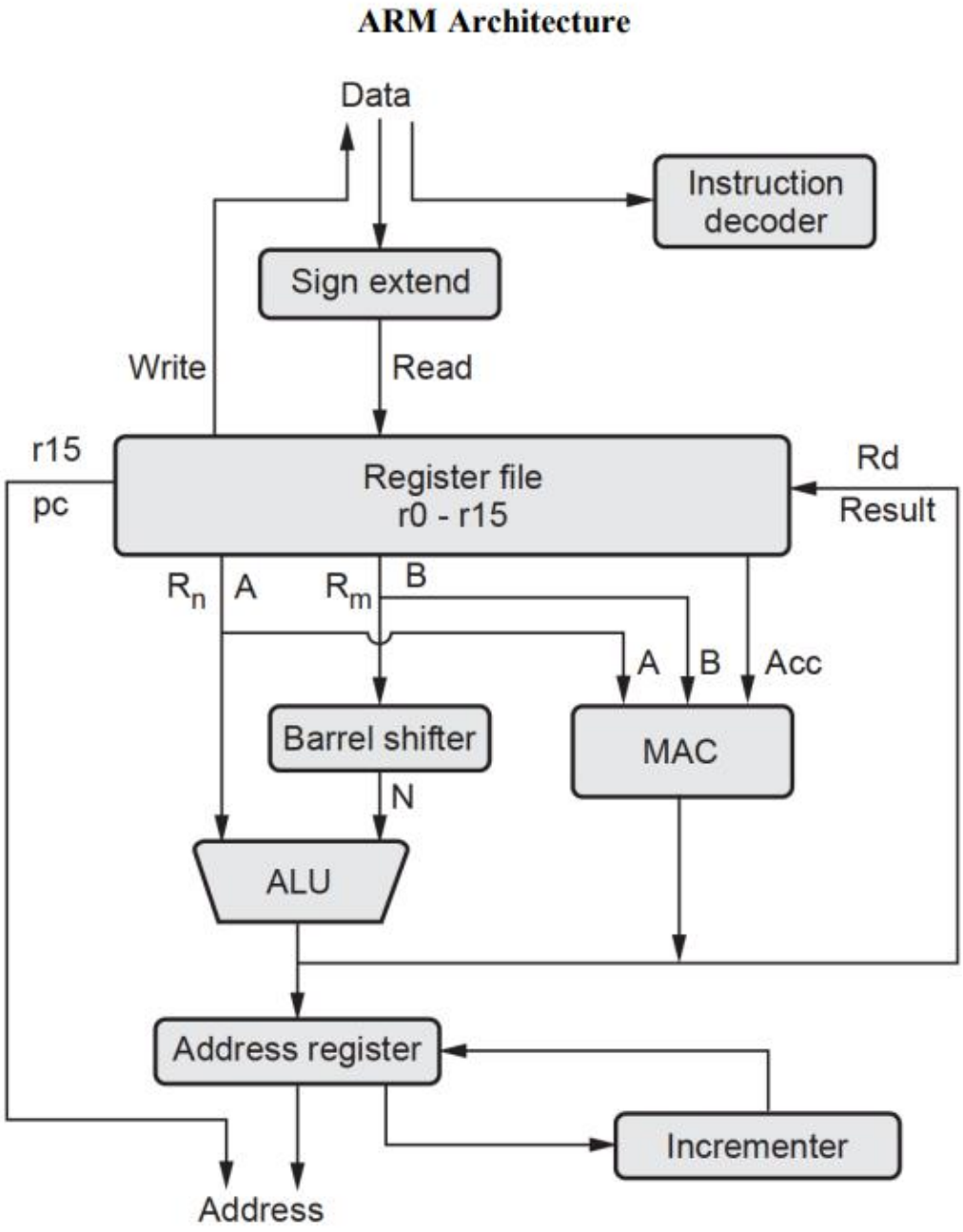
Memory Organization of Intel 8051



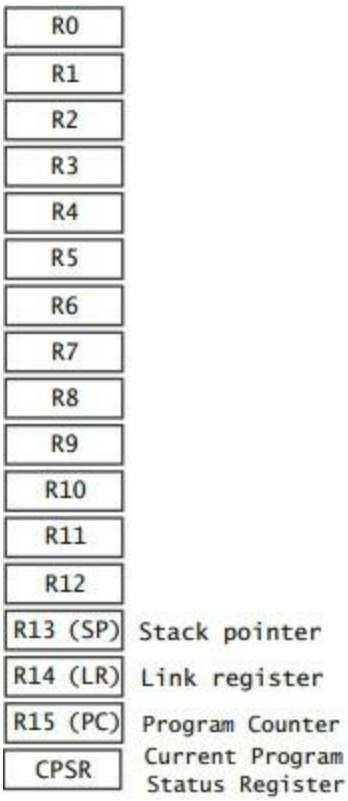
Address Range	Register Bank
00H to 07H	Register Bank 0
08H to 0FH	Register Bank 1
10H to 17H	Register Bank 2
18H to 1FH	Register Bank 3



Module 2



ARM Register Organization


R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13 (SP)
Stack pointer
R14 (LR)
Link register
R15 (PC)
Program Counter
CPSR
Current Program
Status Register

CPSR (Current Program Status Register)

- The ARM core uses the ***cpsr*** to monitor and control internal operations.
- The ***cpsr*** is a dedicated 32-bit register and resides in the register file.
- The ***cpsr*** is divided into **four fields, each 8 bits wide**: **flags**, status, extension, and control. In current designs the extension and status fields are reserved for future use. The control field contains the processor mode, state, and interrupt mask bits. The flags field contains the condition flags.

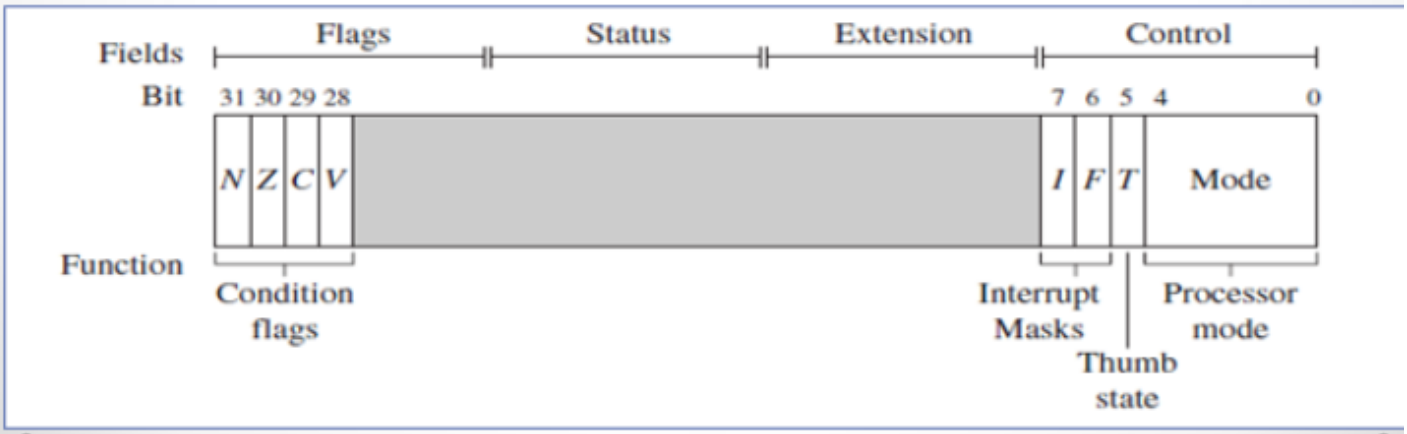
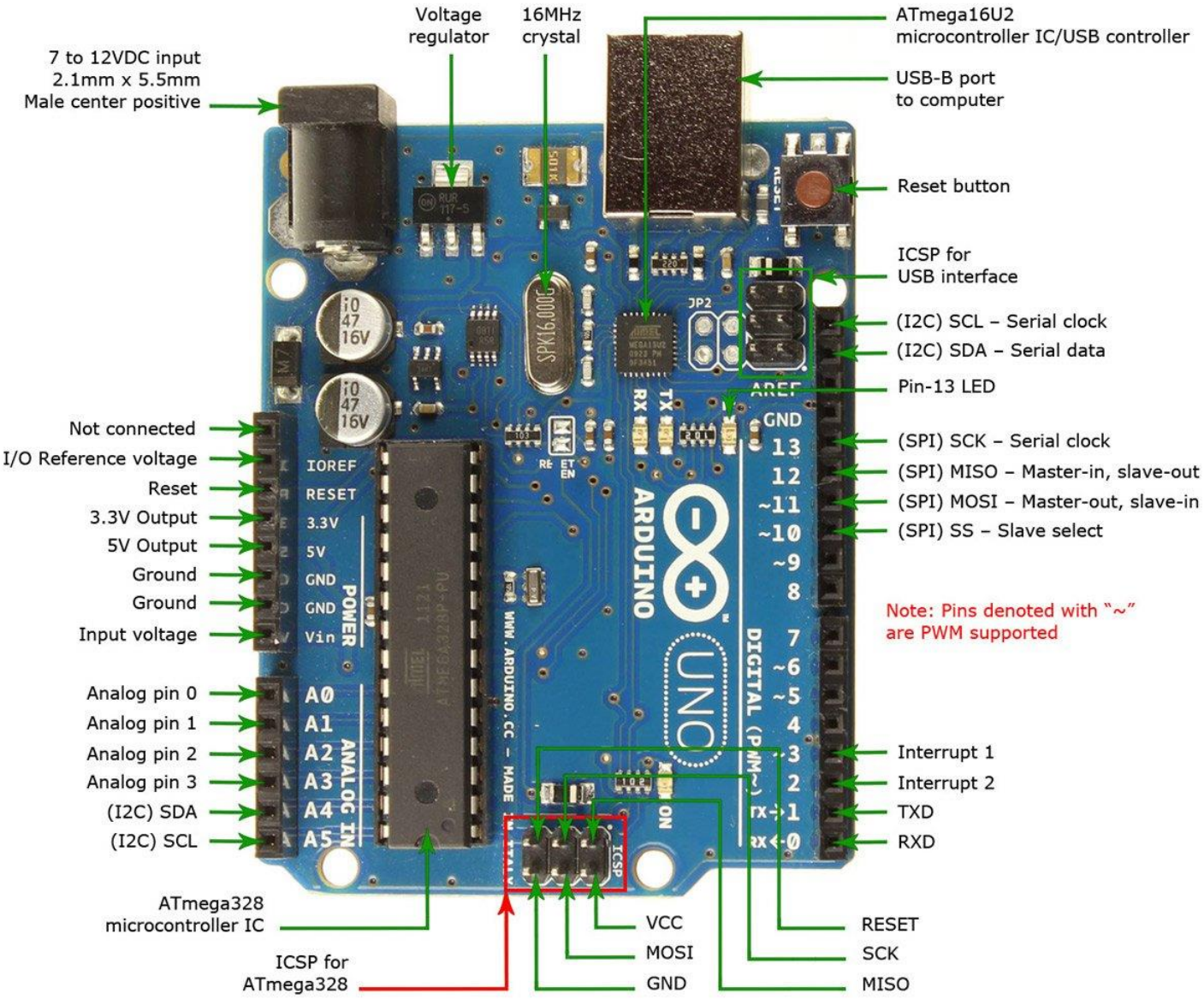
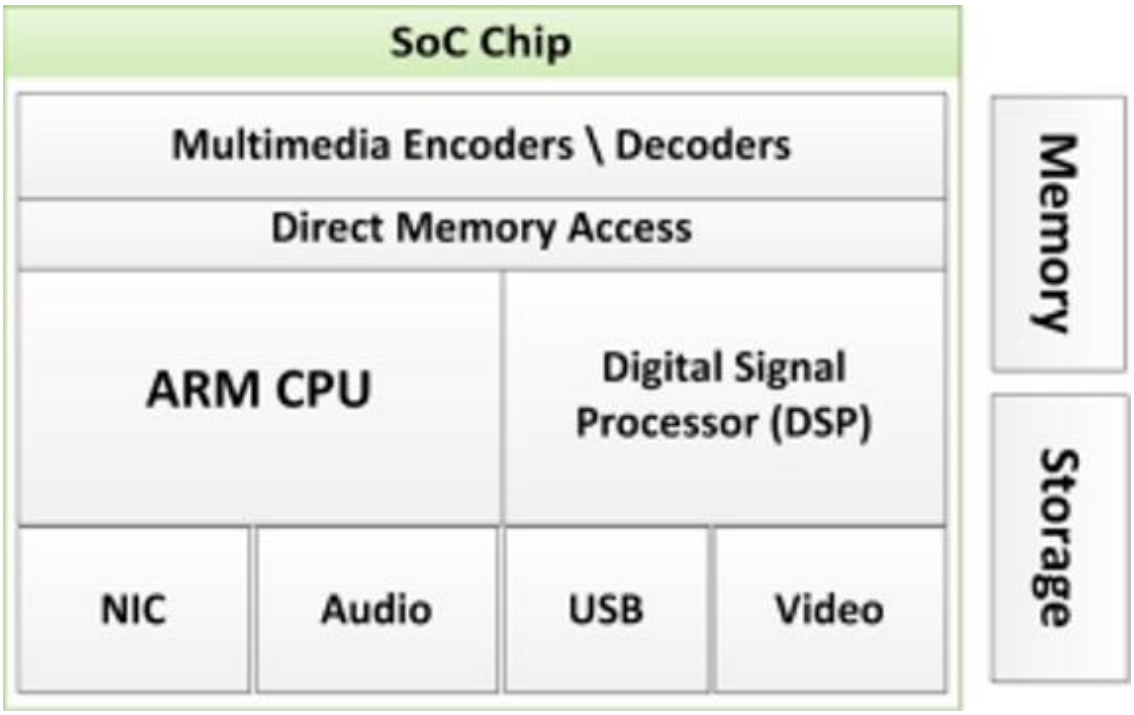


Fig 3: A generic program status register (*psr*).

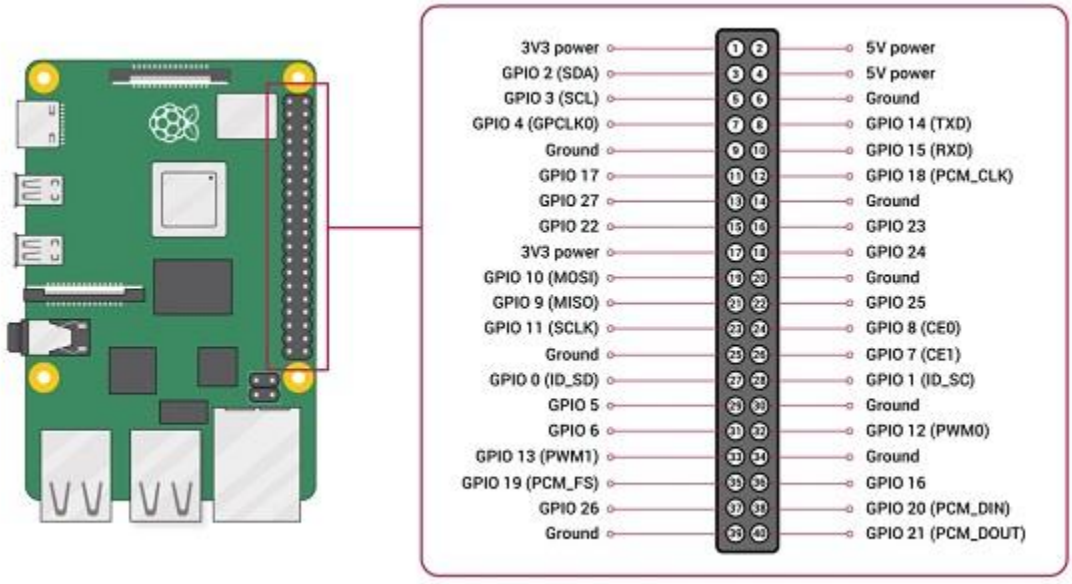
Arduino Uno



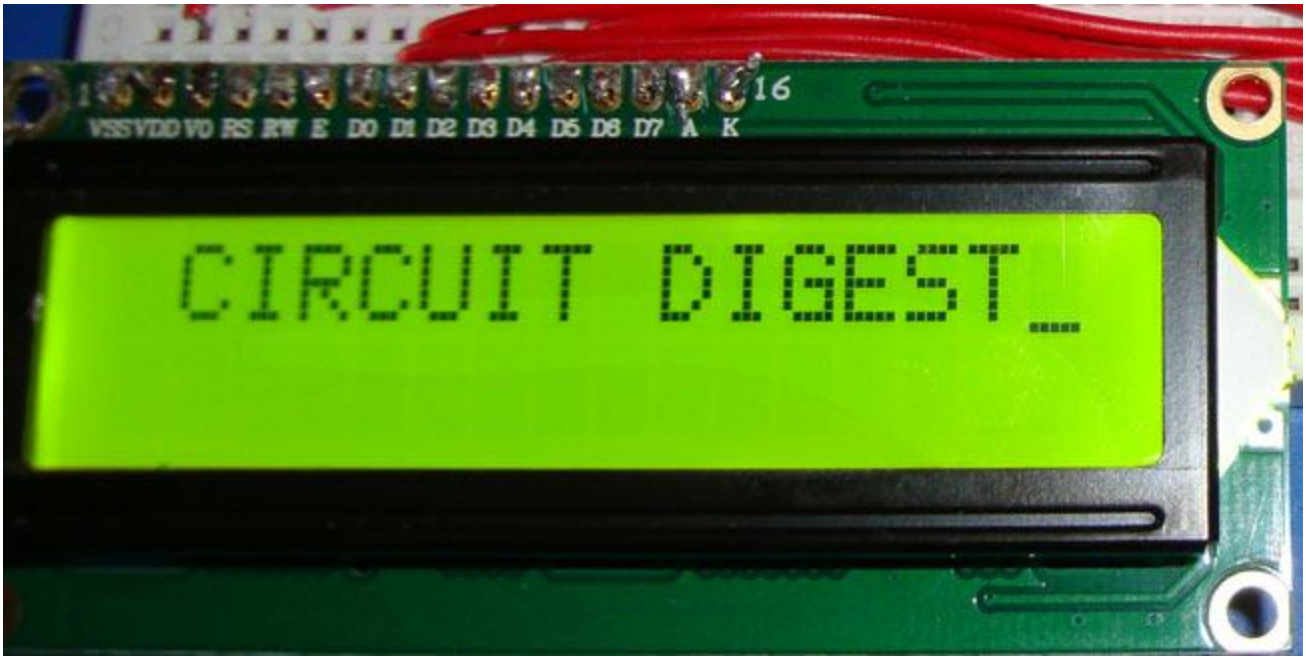
Architecture of SoC System on chips

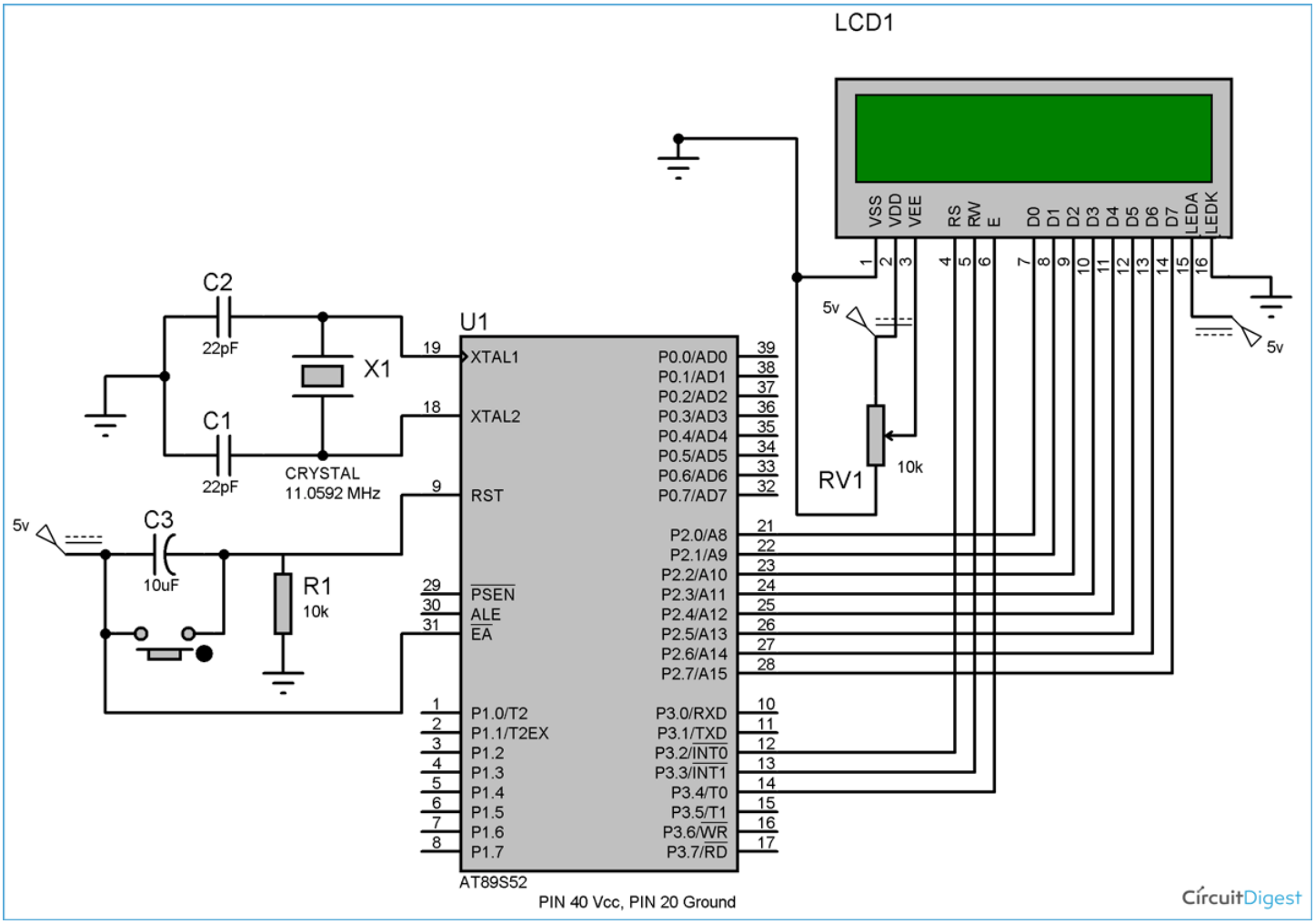


Raspberry Pi



LCD Interfacing with 8051 Microcontroller


VSS VDD VO RS RW E D0 D1 D2 D3 D4 D5 D6 D7 A K
16
CIRCUIT DIGEST


LCD1
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7 LEDA LEDK
5v
5v
RV1
10k
C2
22pF
X1
C1
22pF
CRYSTAL
11.0592 MHz
U1
XTAL1
XTAL2
RST
PSEN
ALE
EA
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P1.0/T2
P1.1/T2EX
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
5v
C3
10uF
R1
10k
AT89S52
PIN 40 Vcc, PIN 20 Ground
CircuitDigest

Module 3

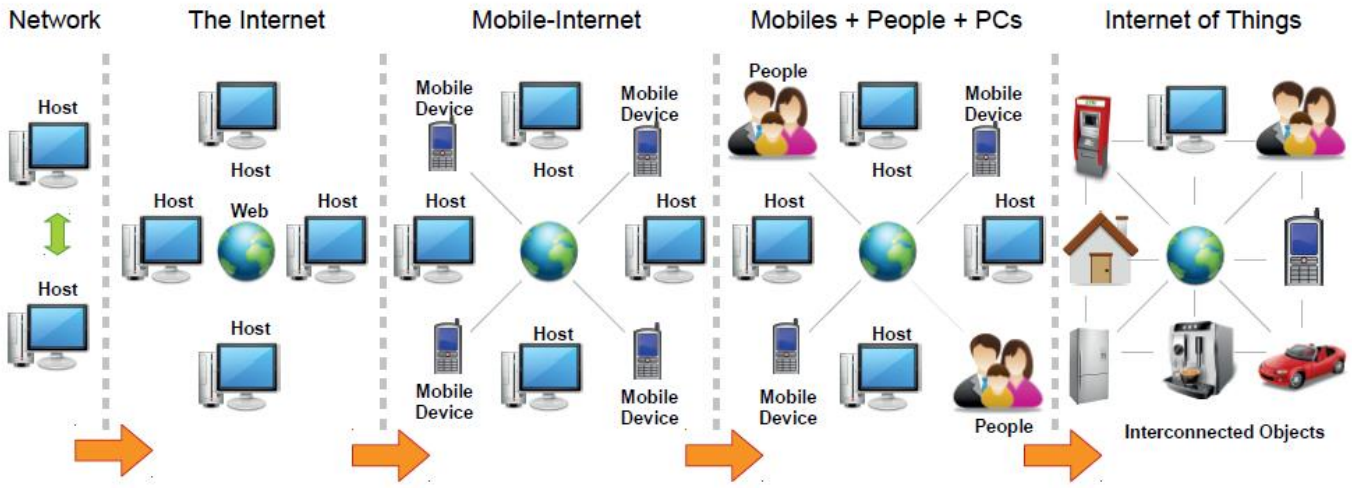
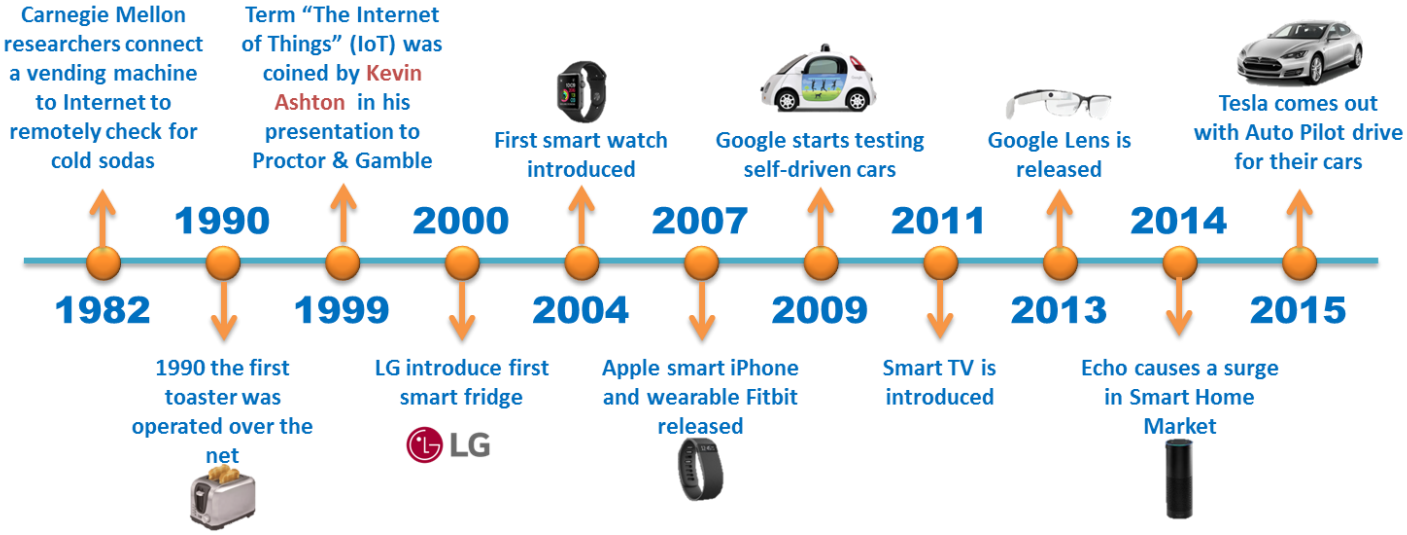
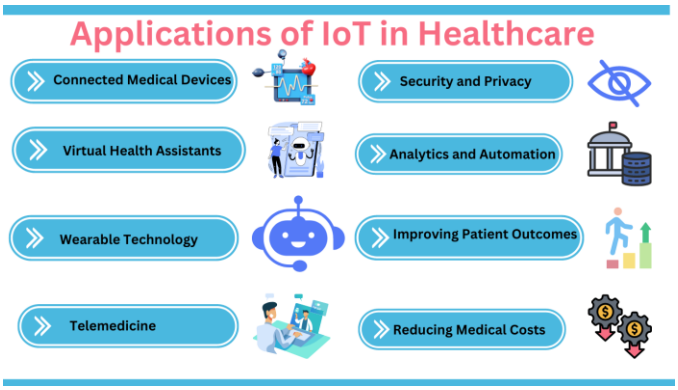
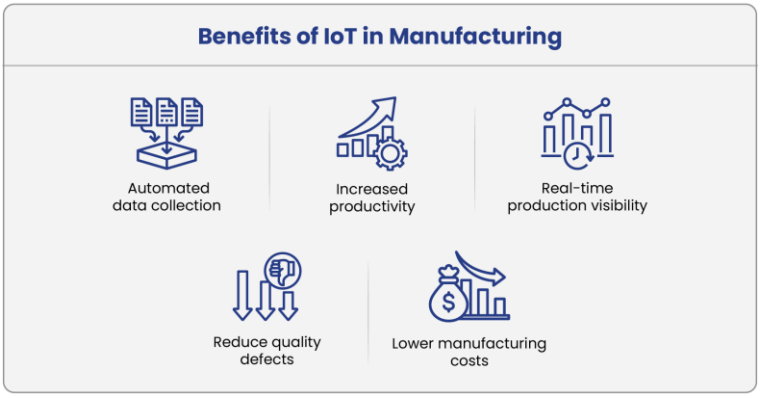


Fig. 1. Evolution of the Internet in five phases. The evolution of Internet begins with connecting two computers together and then moved towards creating World Wide Web by connecting large number of computers together. The mobile-Internet emerged by connecting mobile devices to the Internet. Then, peoples' identities joined the Internet via social networks. Finally, it is moving towards Internet of Things by connecting every day objects to the Internet.







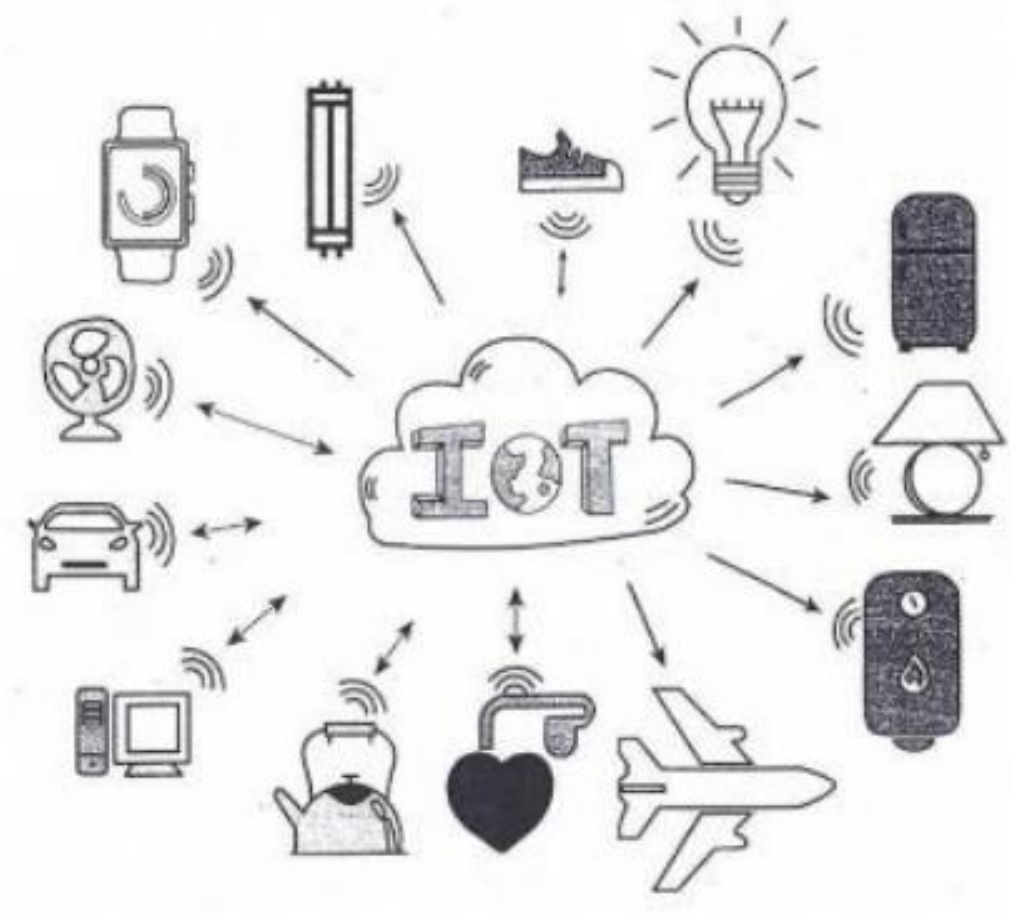
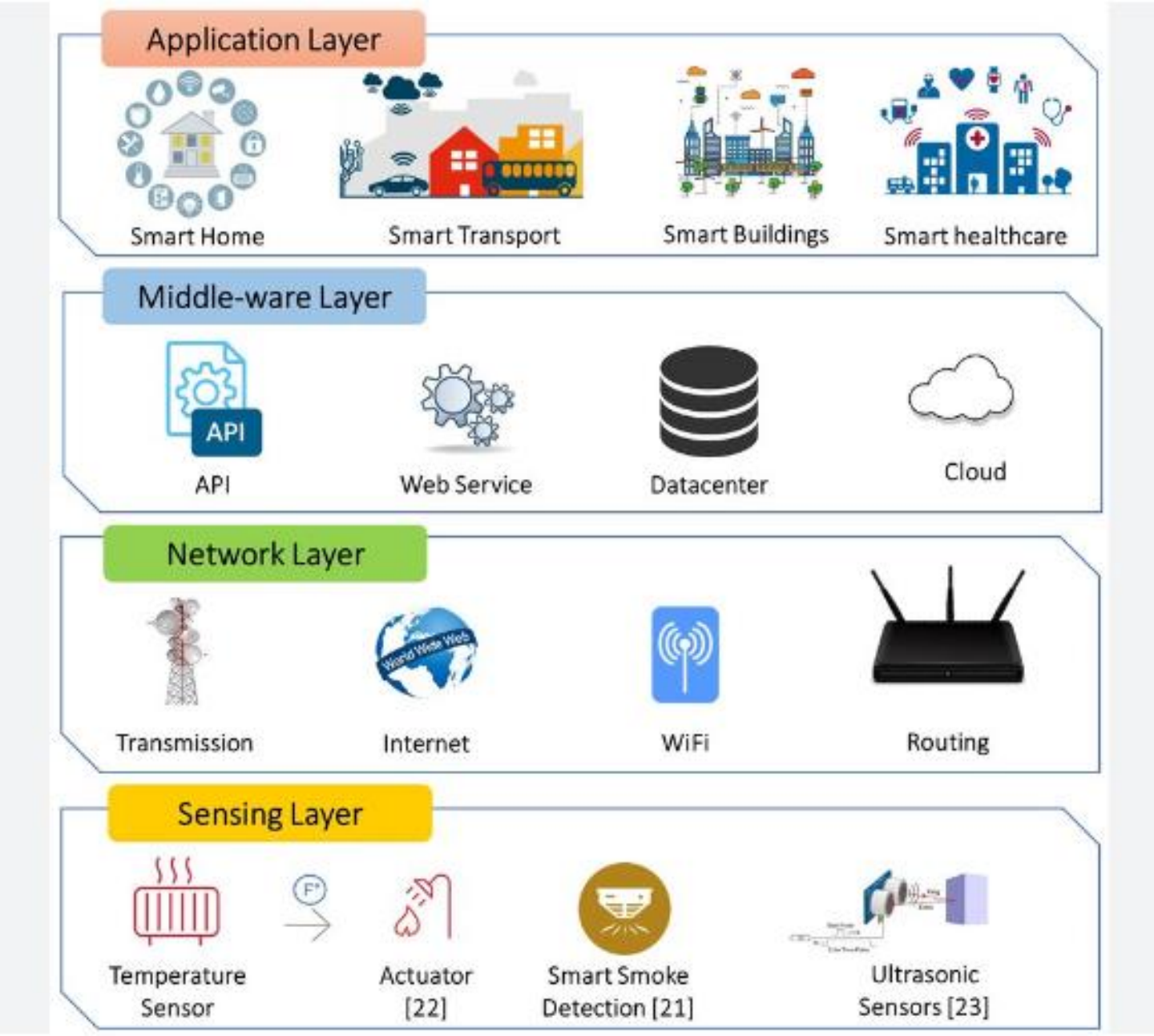


Figure 1.6 "Things" in internet of things


Application Layer
Smart Home
Smart Transport
Smart Buildings
Smart healthcare
Middle-ware Layer
API
API
Web Service
Datacenter
Cloud
Network Layer
Transmission
Internet
WiFi
Routing
Sensing Layer
Temperature Sensor
Actuator [22]
Smart Smoke Detection [21]
Ultrasonic Sensors [23]

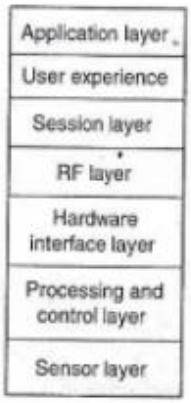
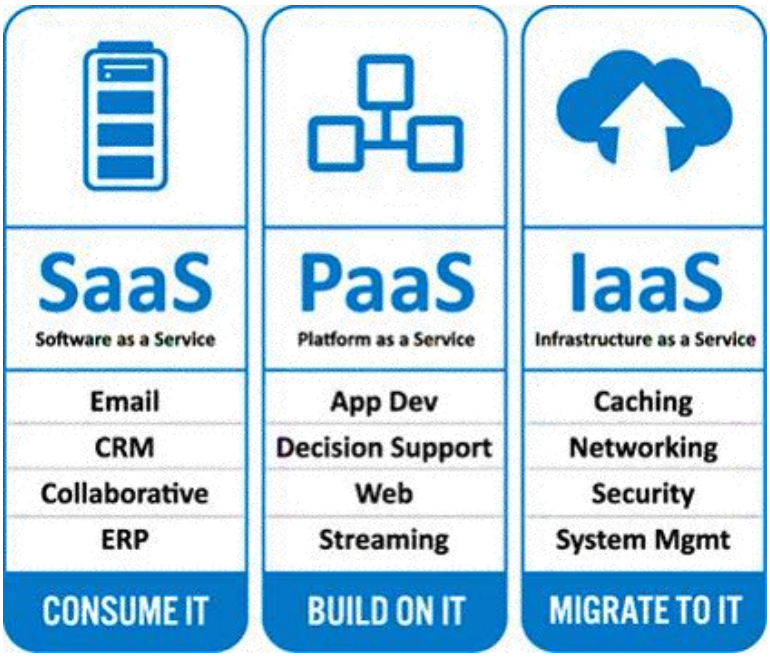
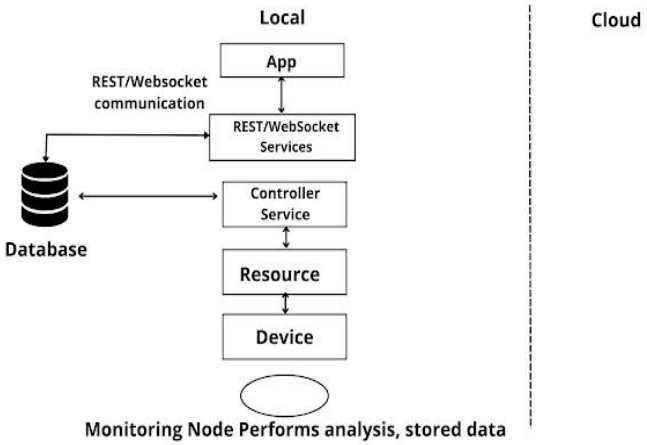


Figure 1.16 IoT layers

Figure 1.17 Sensors that act as enabling technologies





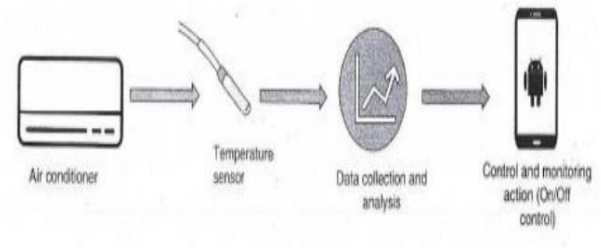


Figure 1.19 Level 1-IoT application

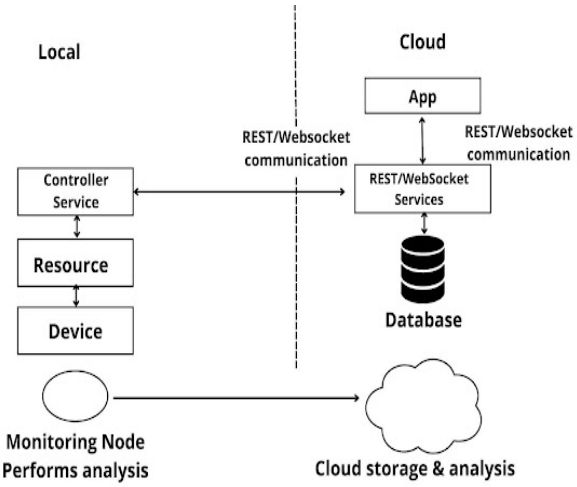
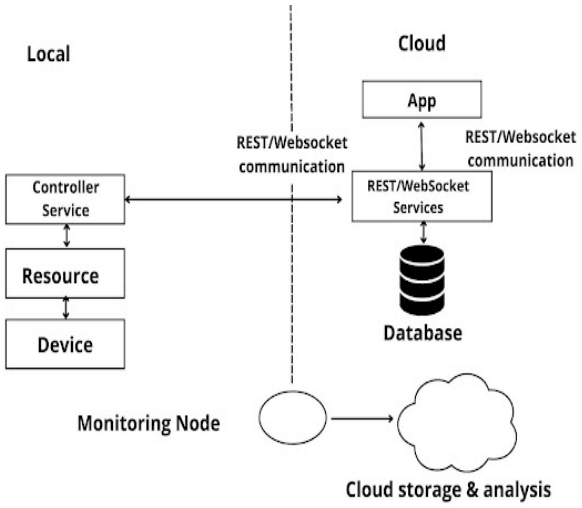


Figure 1.20 Level 2-to-IoT application



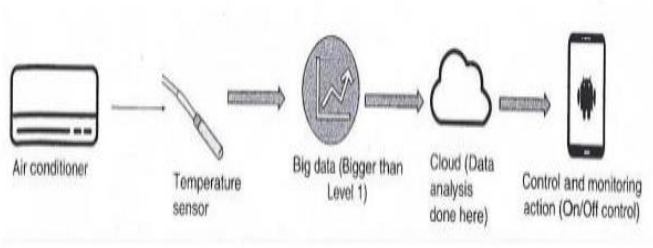
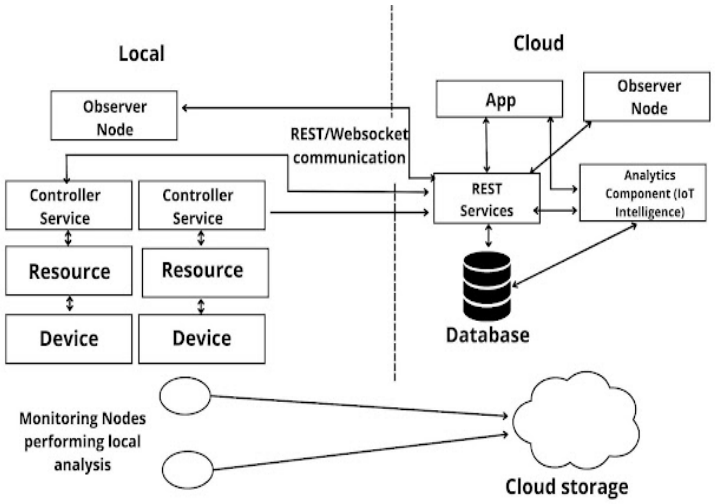


Figure 1.21 Level 3-IoT application



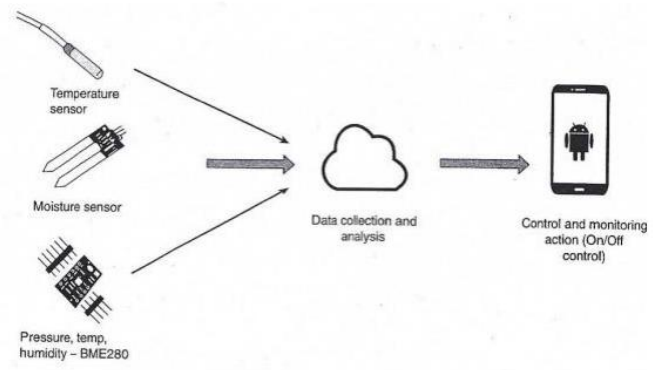
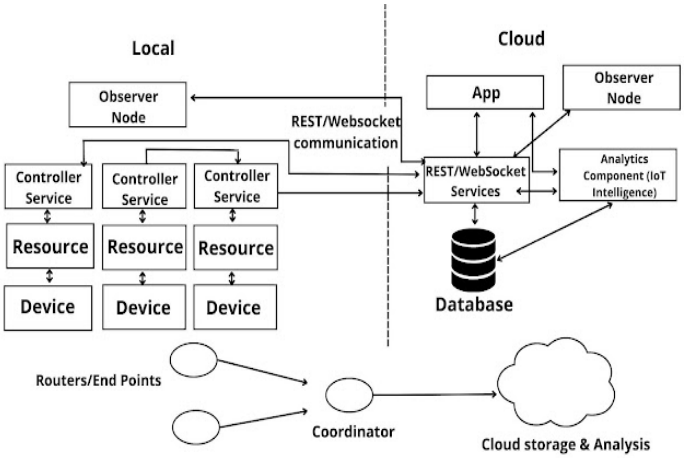


Figure 1.22 Level 4-IoT application



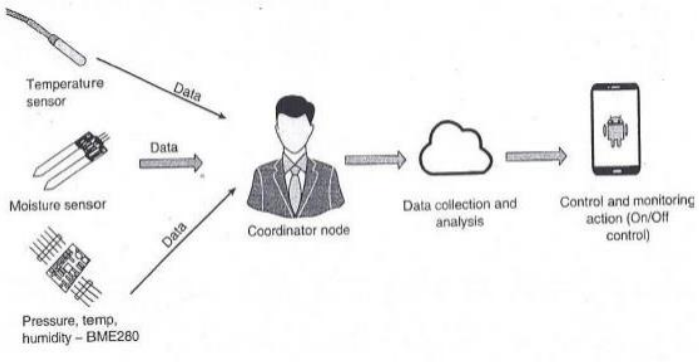
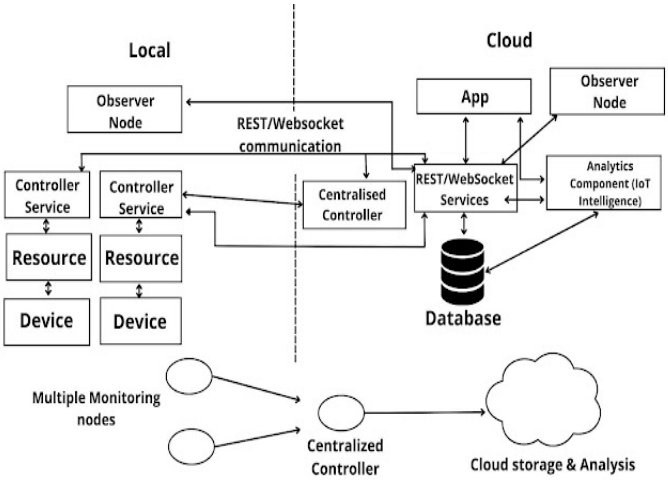
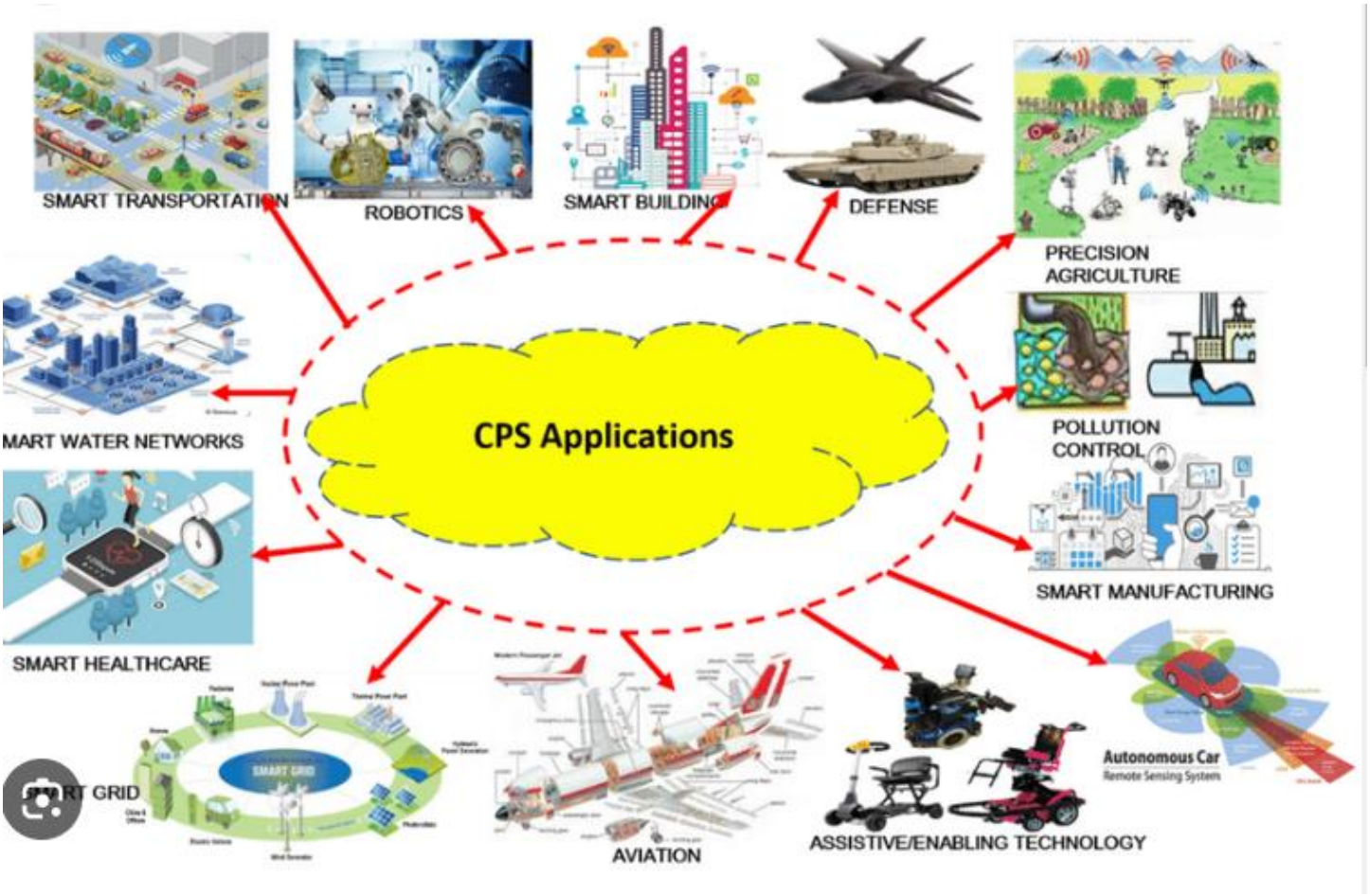
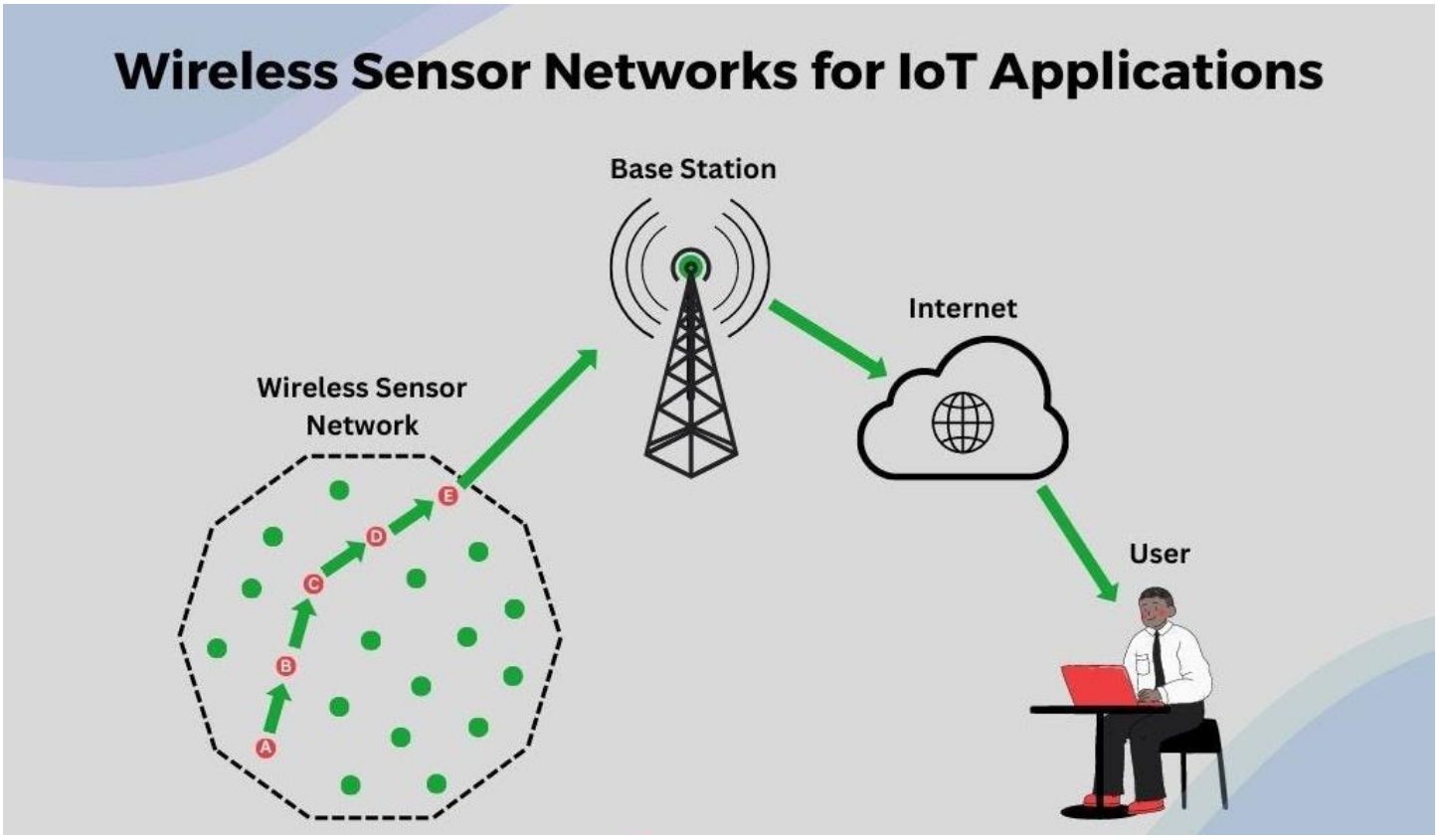


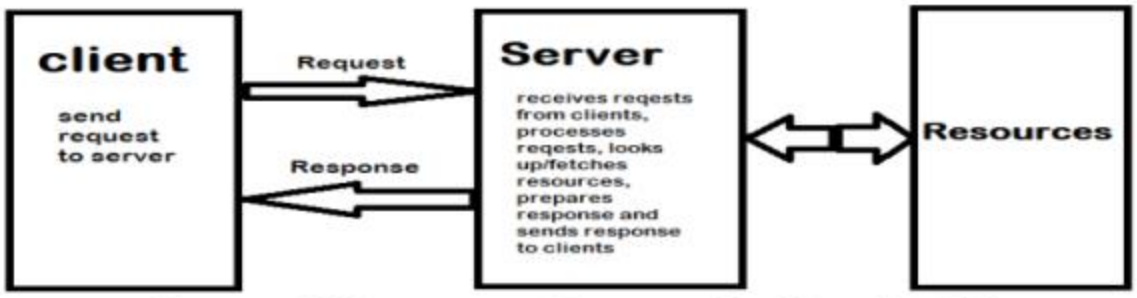
Figure 1.23 Level 5-IoT application



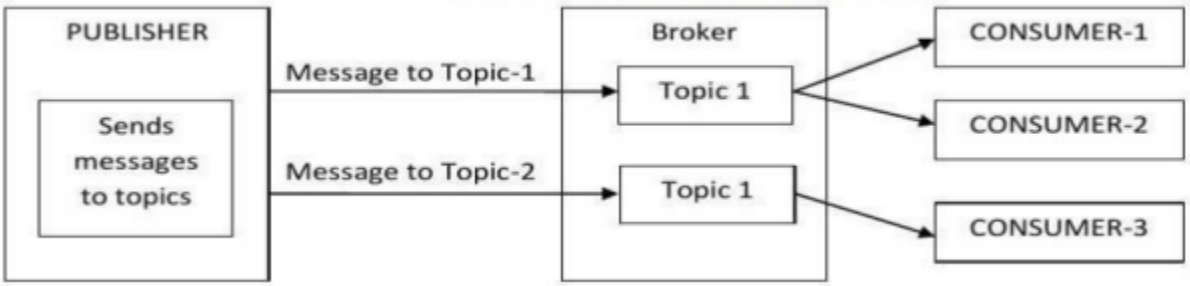
Cyber Physical System

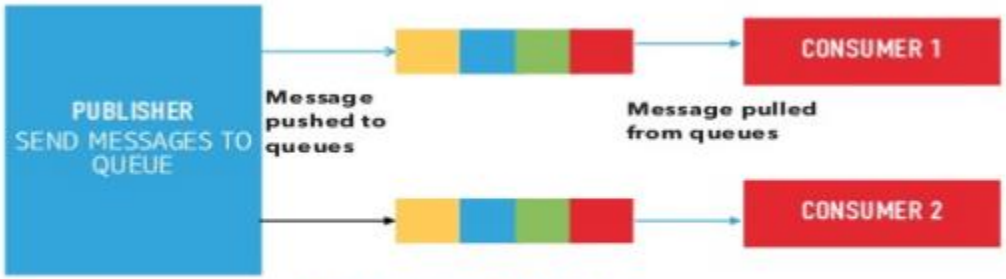

SMART TRANSPORTATION
ROBOTICS
SMART BUILDING
DEFENSE
PRECISION AGRICULTURE
CPS Applications
MART WATER NETWORKS
POLLUTION CONTROL
SMART MANUFACTURING
SMART HEALTHCARE
RT GRID
AVIATION
ASSISTIVE/ENABLING TECHNOLOGY
Autonomous Car
Remote Sensing System


Wireless Sensor Networks for IoT Applications
Base Station
Internet
Wireless Sensor Network
User
A
B
C
D
E

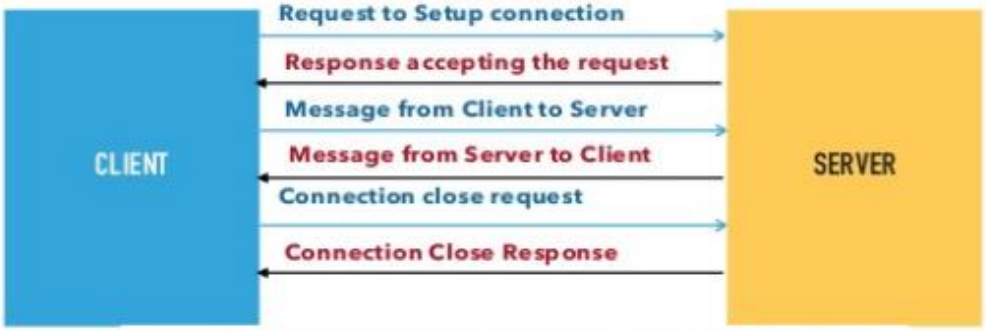


Request-Response Communication Model

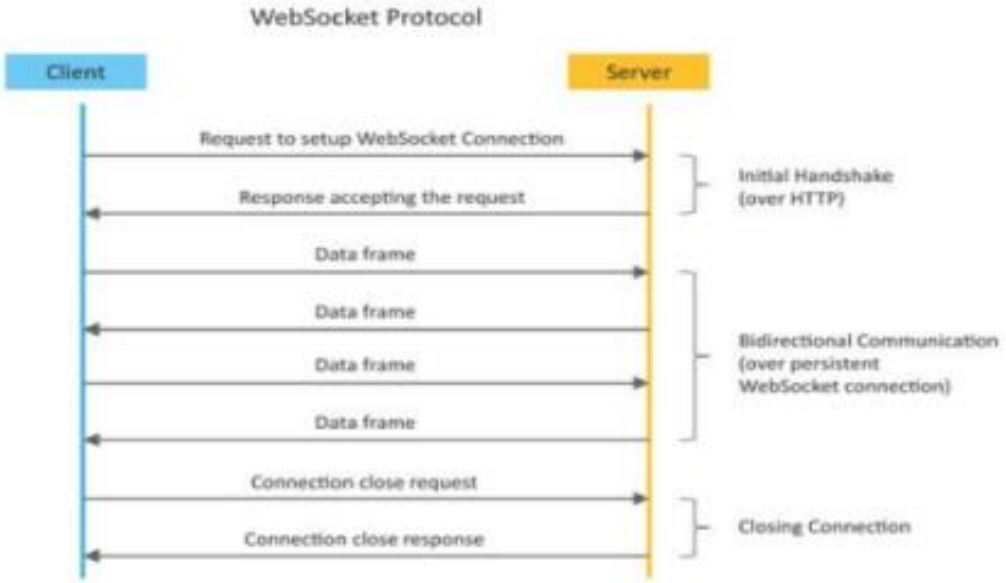




PUSH PULL MODEL



EXCLUSIVE PAIR COMMUNICATION MODEL


WebSocket Protocol
Client
Server
Request to setup WebSocket Connection
Response accepting the request
Initial Handshake
(over HTTP)
Data frame
Data frame
Data frame
Data frame
Bidirectional Communication
(over persistent
WebSocket connection)
Connection close request
Connection close response
Closing Connection

Module 4

Figure 3-2 *Sensors in a Smart Phone*

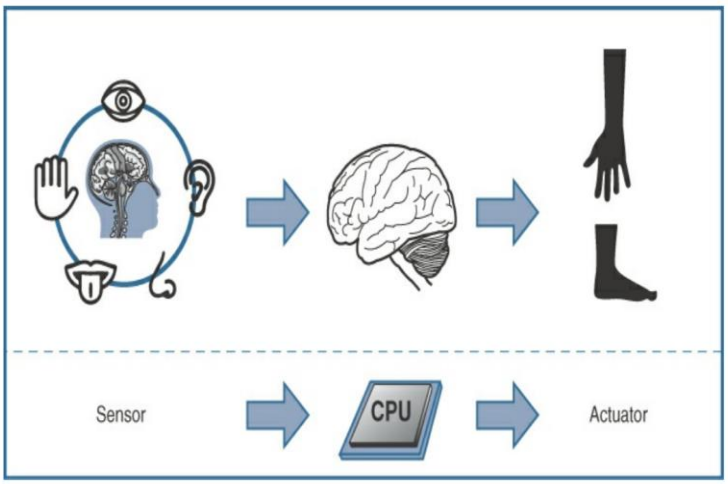
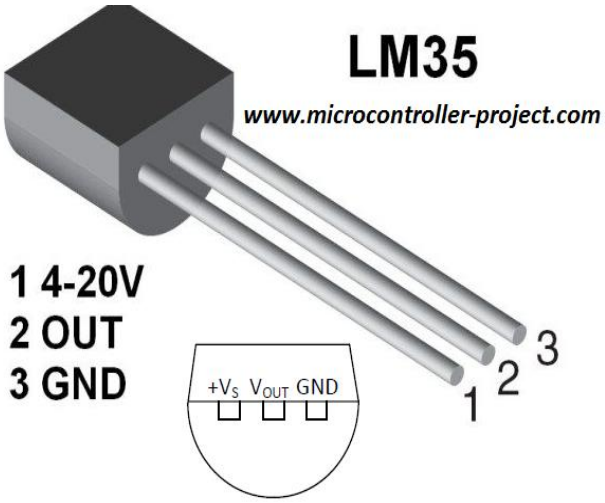
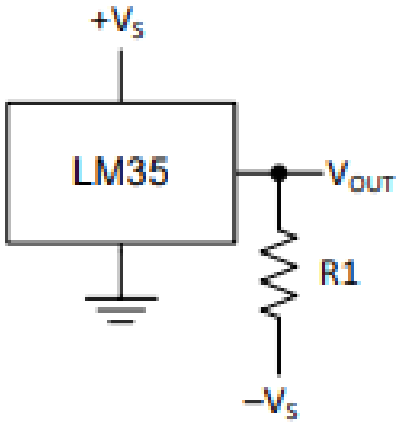


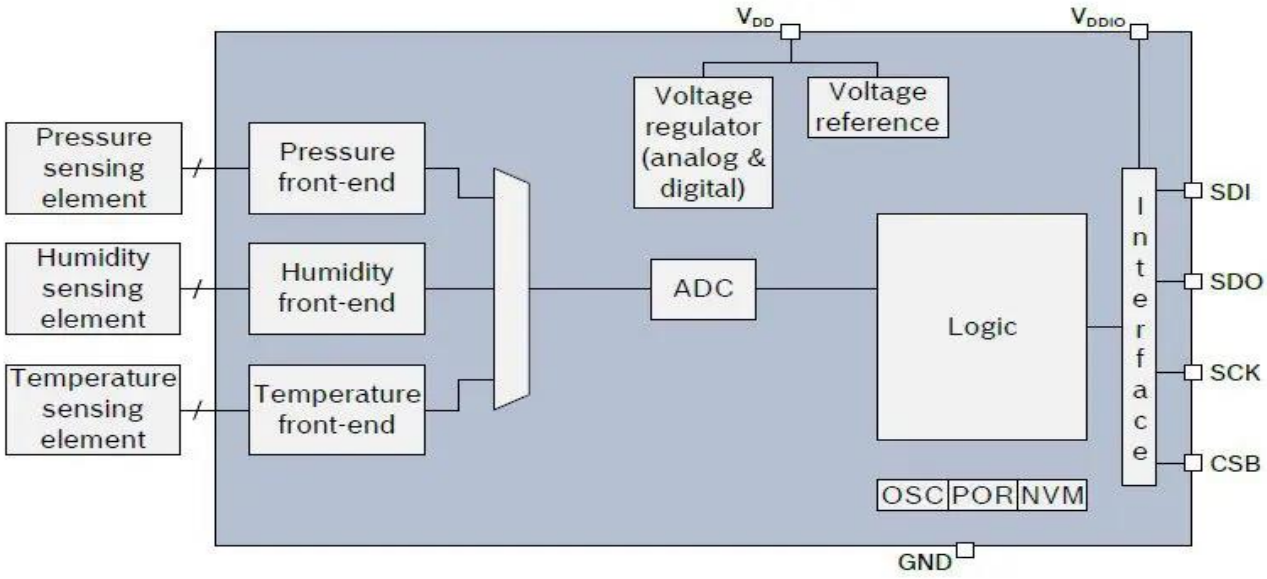
Figure 3-5 *Comparison of Sensor and Actuator Functionality with Humans*



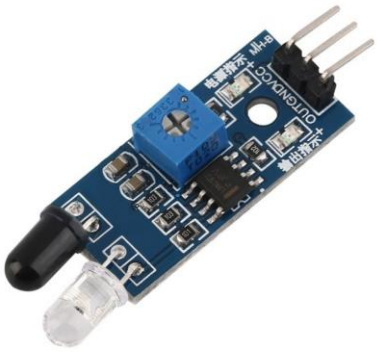
Full-Range Centigrade Temperature Sensor

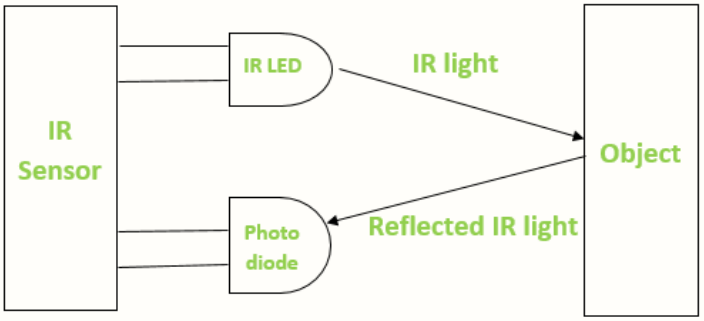


Choose R_1 = –V_S / 50 µA
V_{OUT} = 1500 mV at 150°C
V_{OUT} = 250 mV at 25°C
V_{OUT} = –550 mV at –55°C

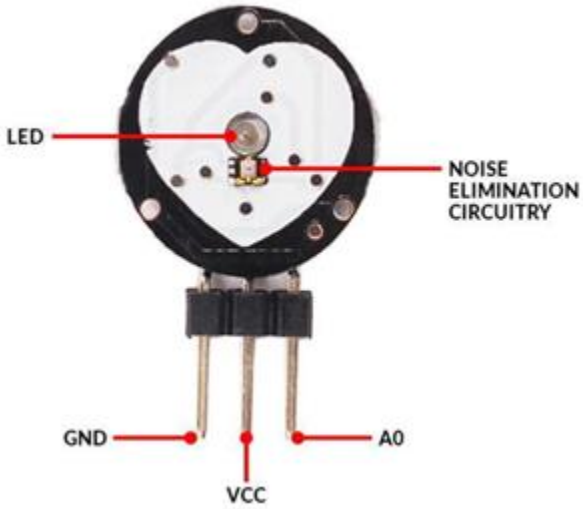


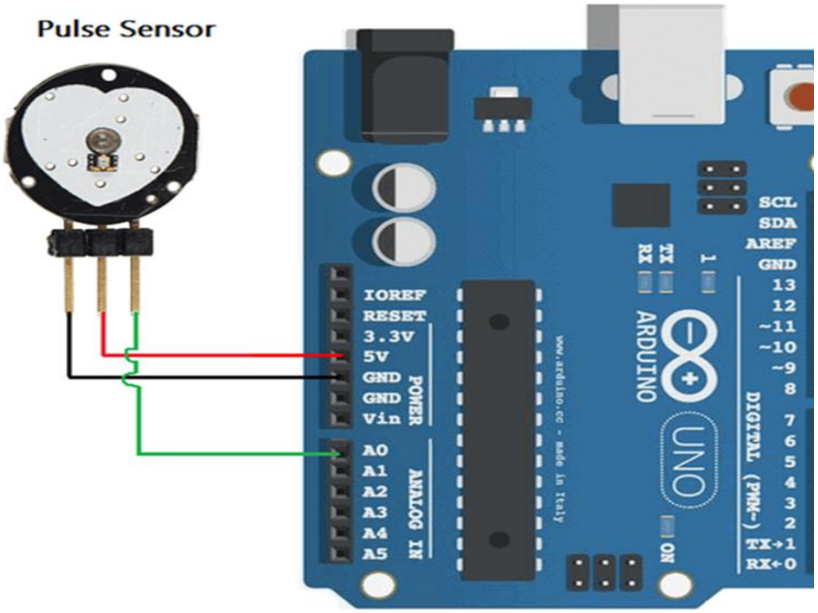
Gas sensor MQ-2

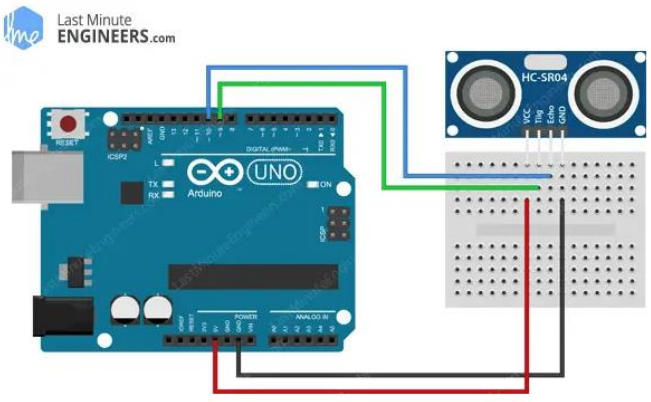




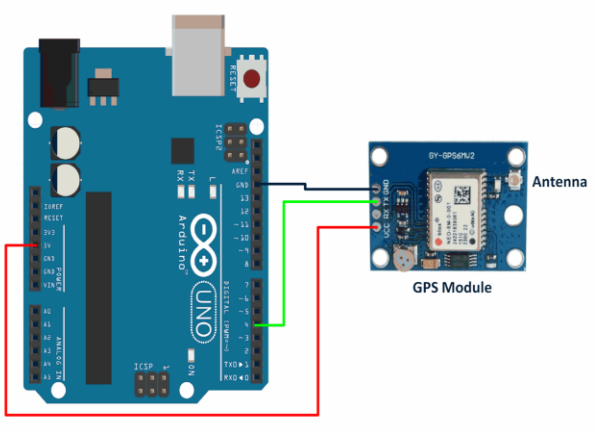
Obstacle sensor

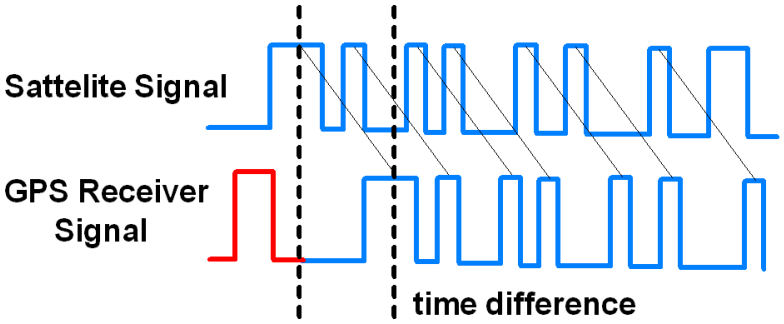


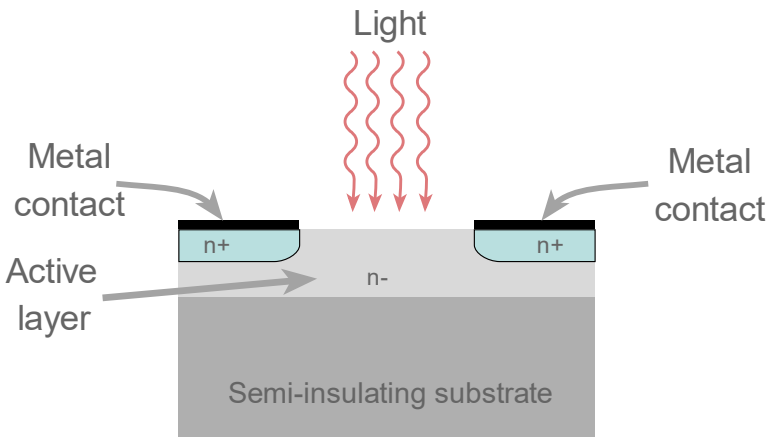




Wiring HC-SR04 Ultrasonic Sensor to Arduino UNO – Normal Mode

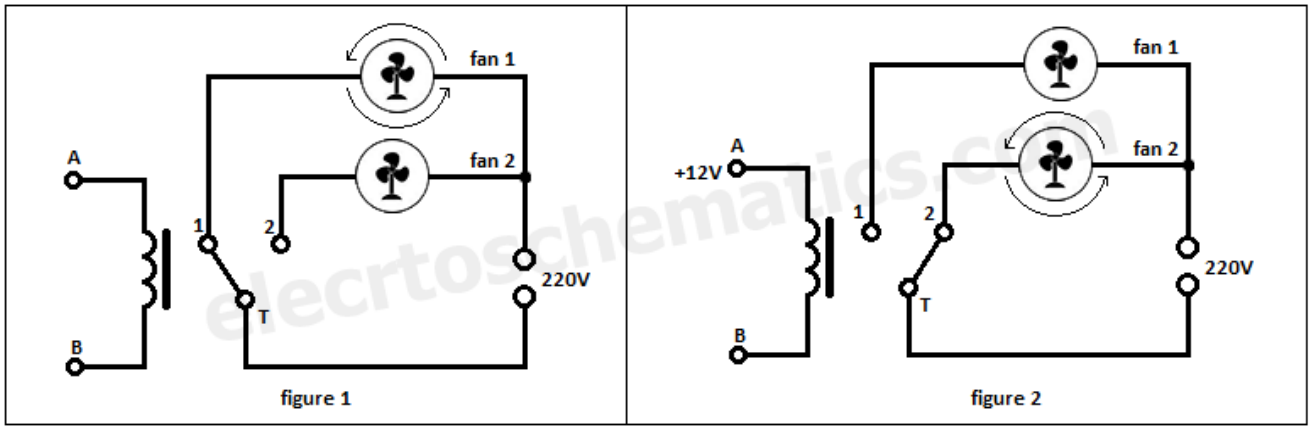




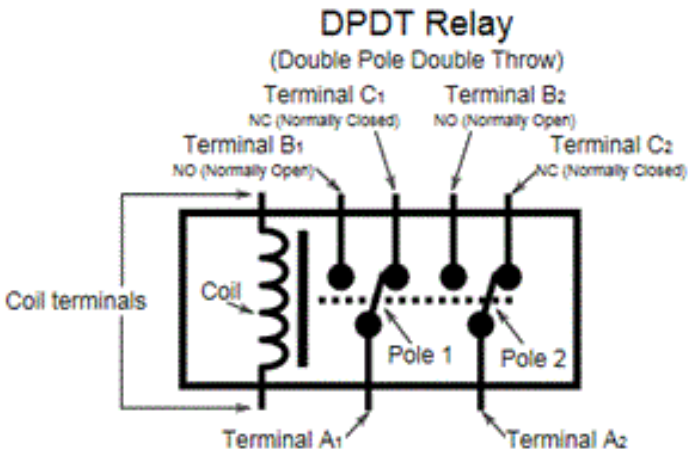


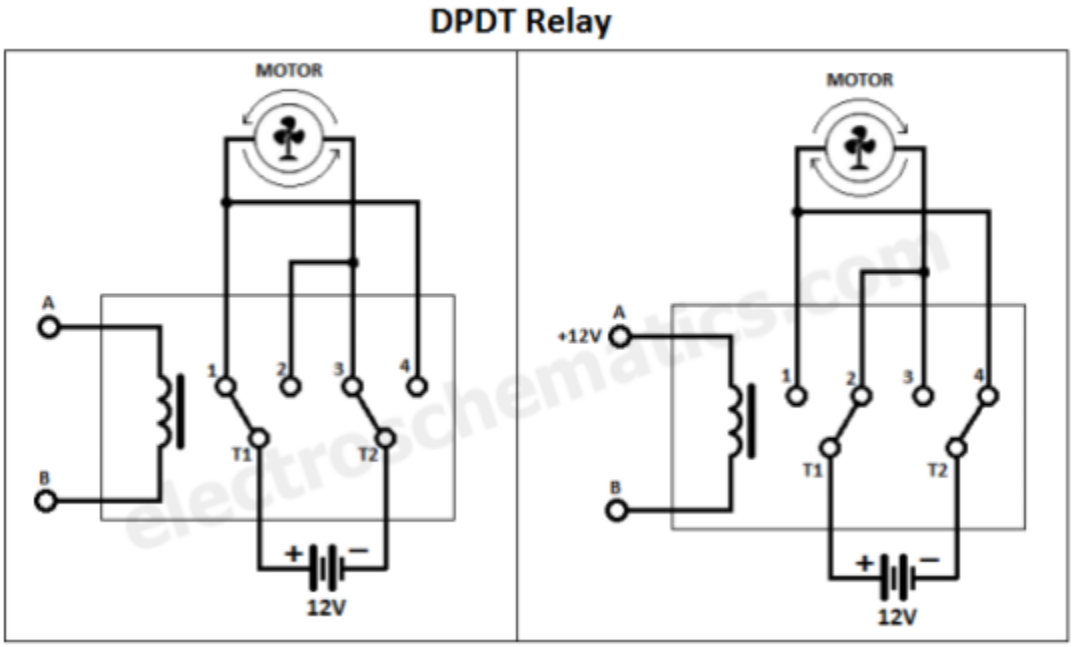
LDR

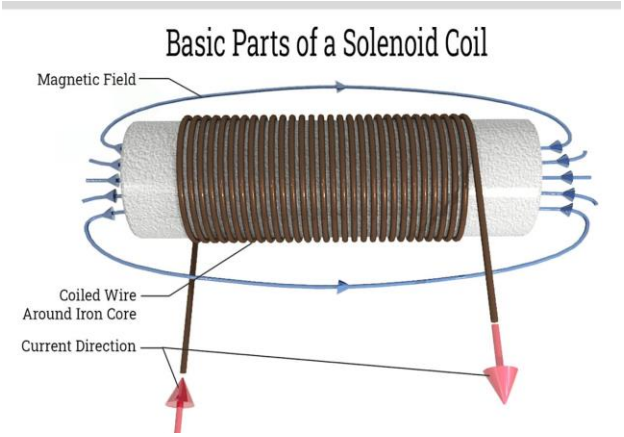
SPDT Relay

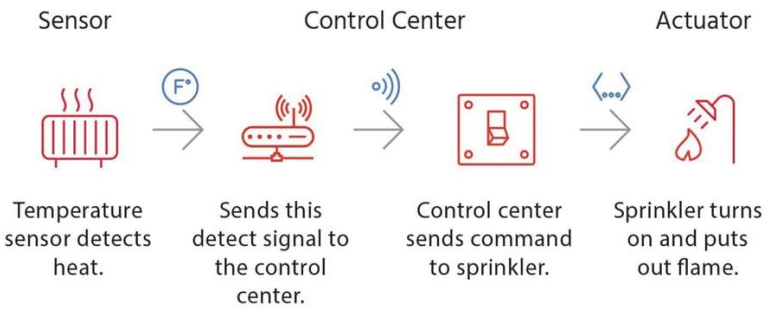


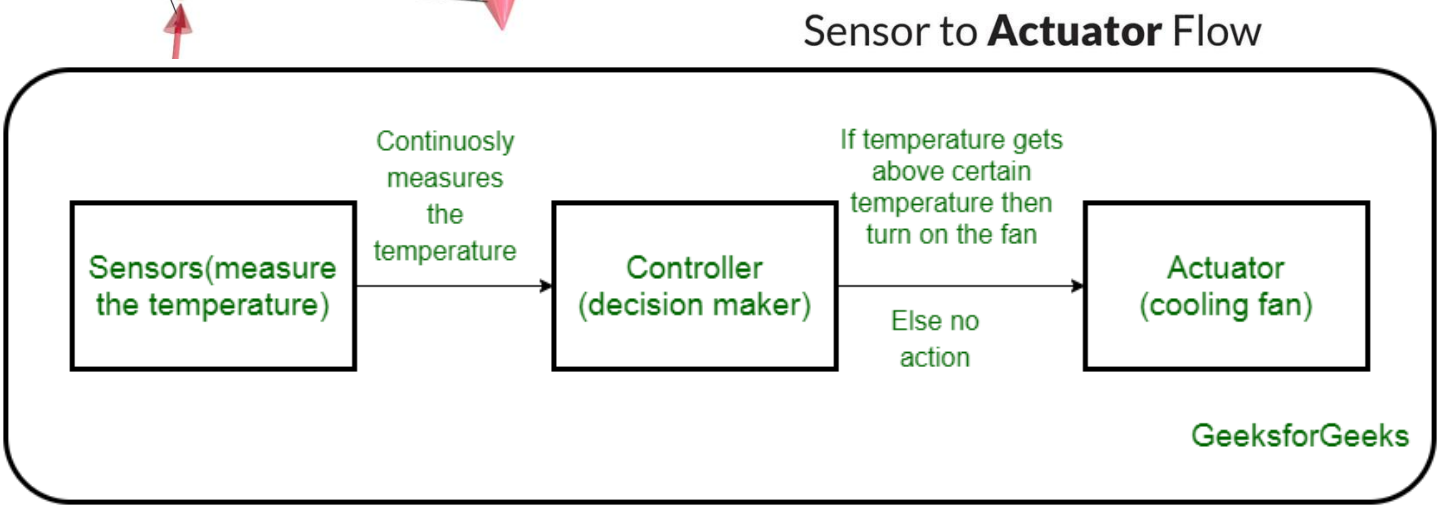

SPDT
Single Pole,
Double Throw
NC
NO
C


DPDT Relay
(Double Pole Double Throw)
Terminal C1
NC (Normally Closed)
Terminal B2
NO (Normally Open)
Terminal B1
NO (Normally Open)
Terminal C2
NC (Normally Closed)
Coil terminals
Coil
Pole 1
Pole 2
Terminal A1
Terminal A2

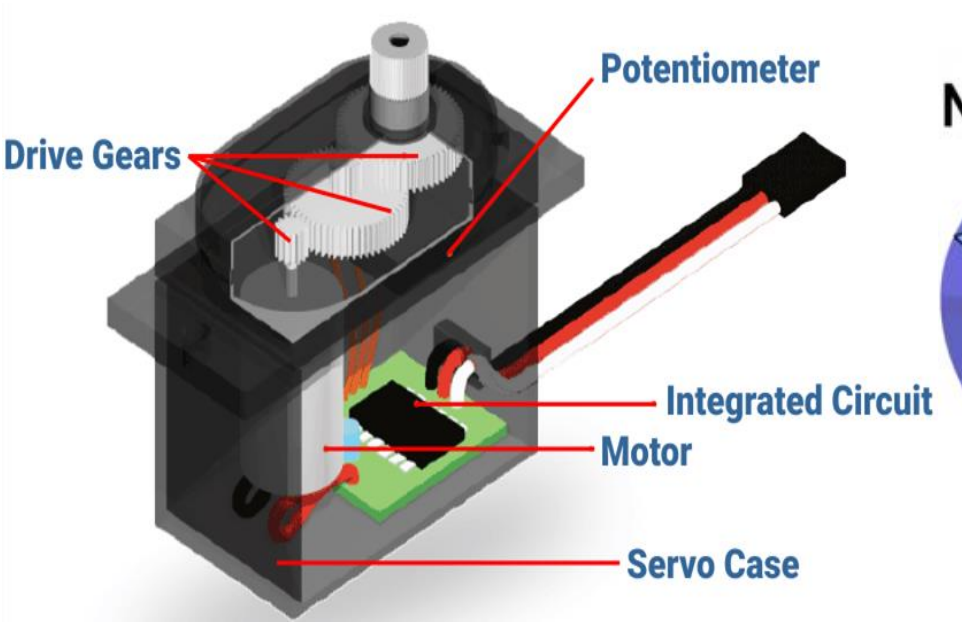

DPDT Relay
MOTOR
A
B
1
2
3
4
T1
T2
+
−
12V
MOTOR
A
+12V
B
1
2
3
4
T1
T2
+
−
12V

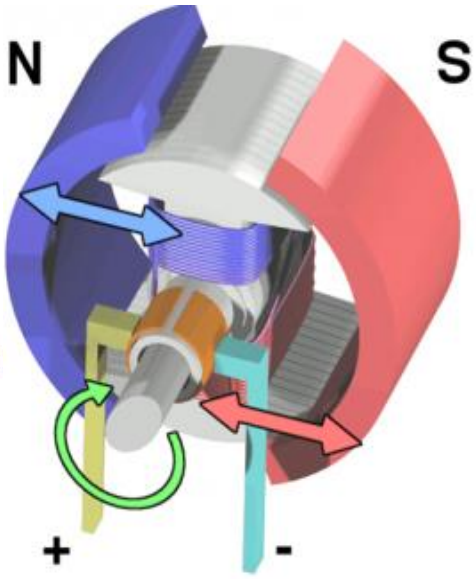

Basic Parts of a Solenoid Coil
Magnetic Field
Coiled Wire
Around Iron Core
Current Direction

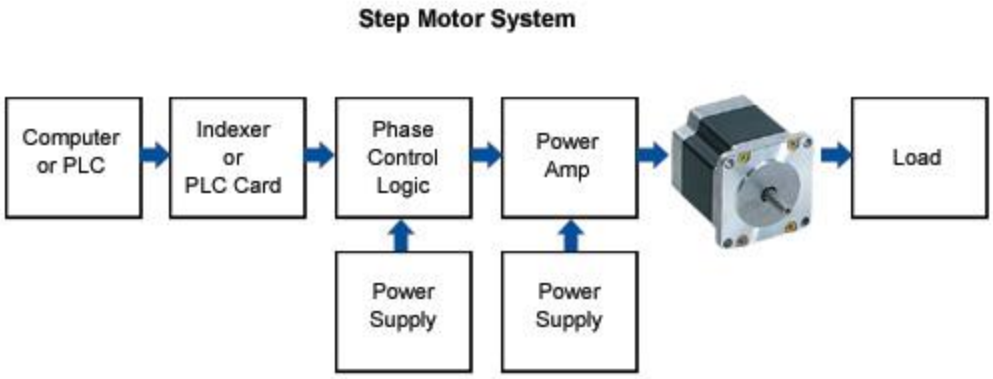

Sensor
Control Center
Actuator
Temperature sensor detects heat.
Sends this detect signal to the control center.
Control center sends command to sprinkler.
Sprinkler turns on and puts out flame.


Sensor to Actuator Flow
Sensors(measure the temperature)
Continuosly measures the temperature
Controller (decision maker)
If temperature gets above certain temperature then turn on the fan
Else no action
Actuator (cooling fan)
GeeksforGeeks

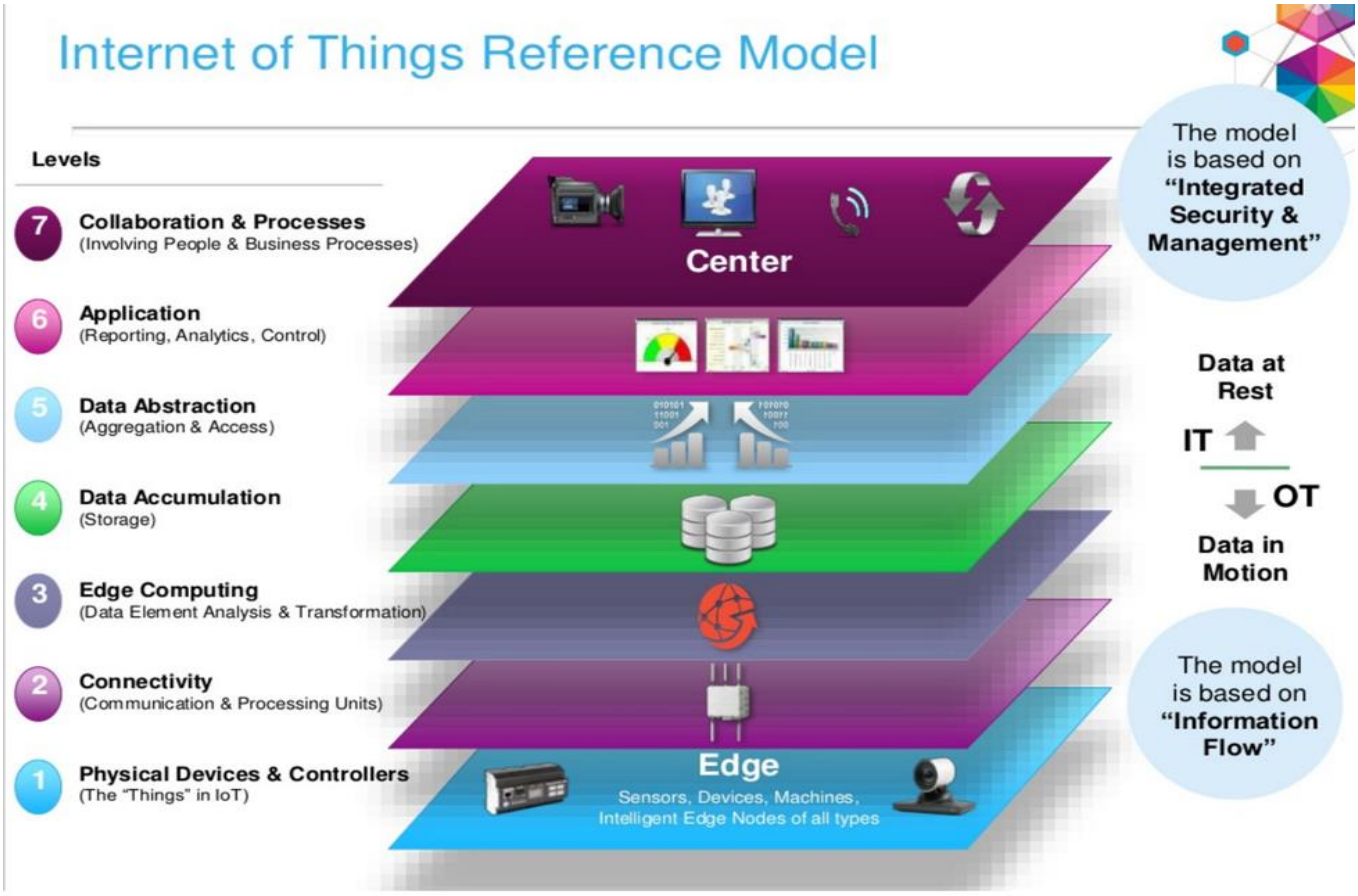
Servo Motor

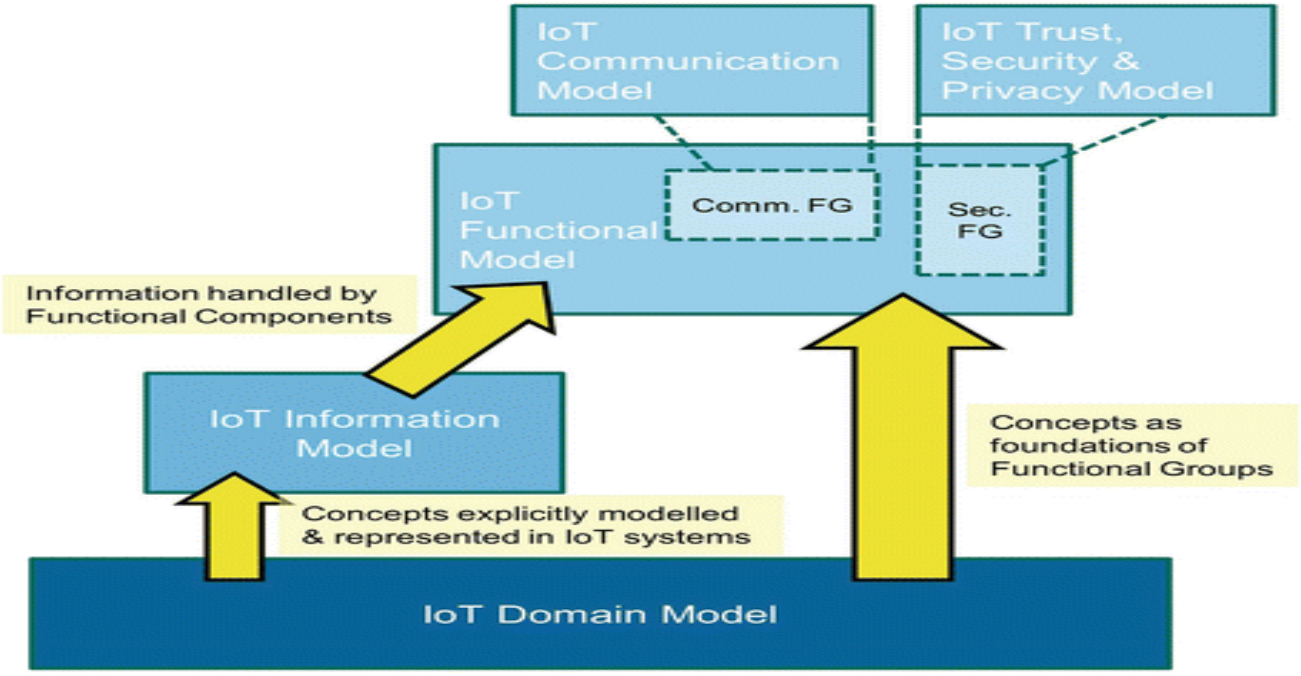

Potentiometer
Drive Gears
Integrated Circuit
Motor
Servo Case


N
S
+
-

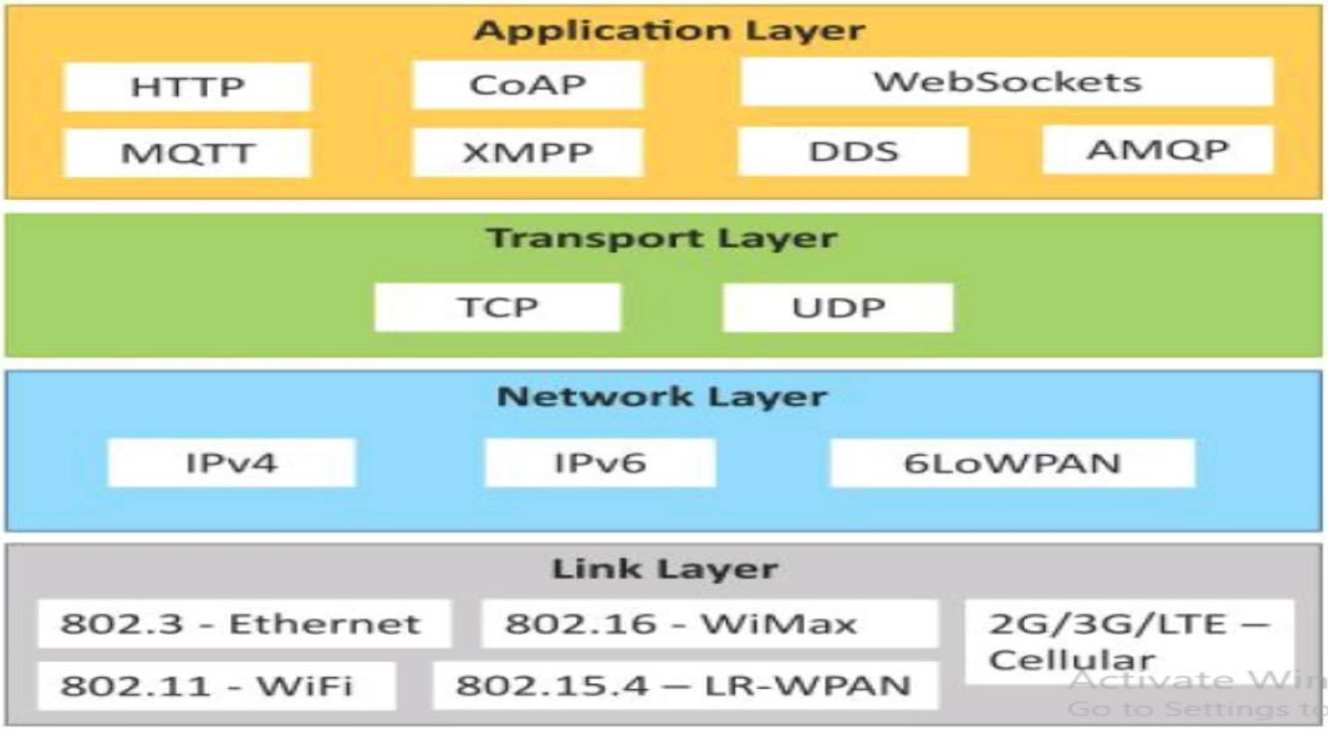

Step Motor System
Computer or PLC
Indexer or PLC Card
Phase Control Logic
Power Amp
Load
Power Supply
Power Supply

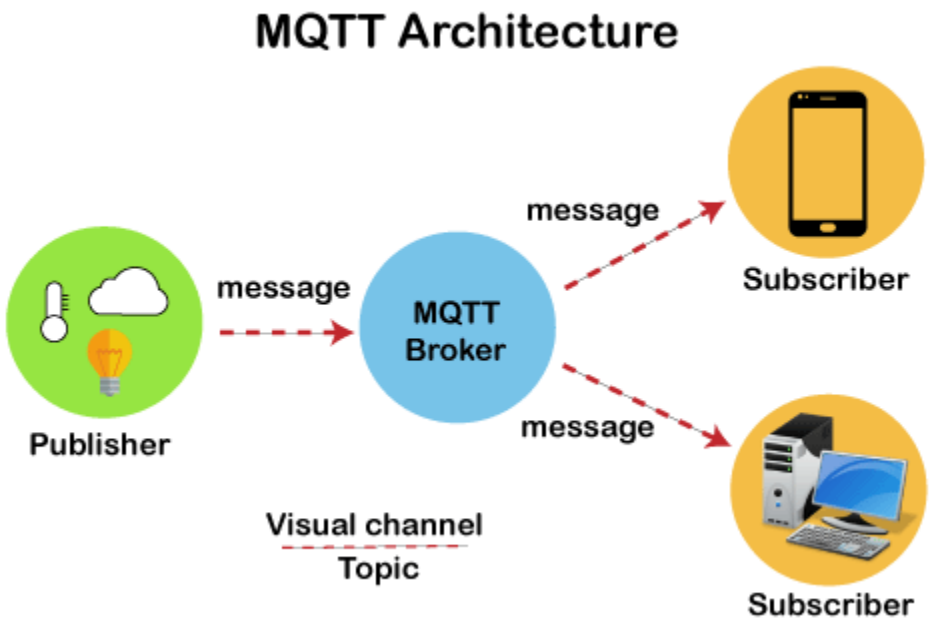
Module 5

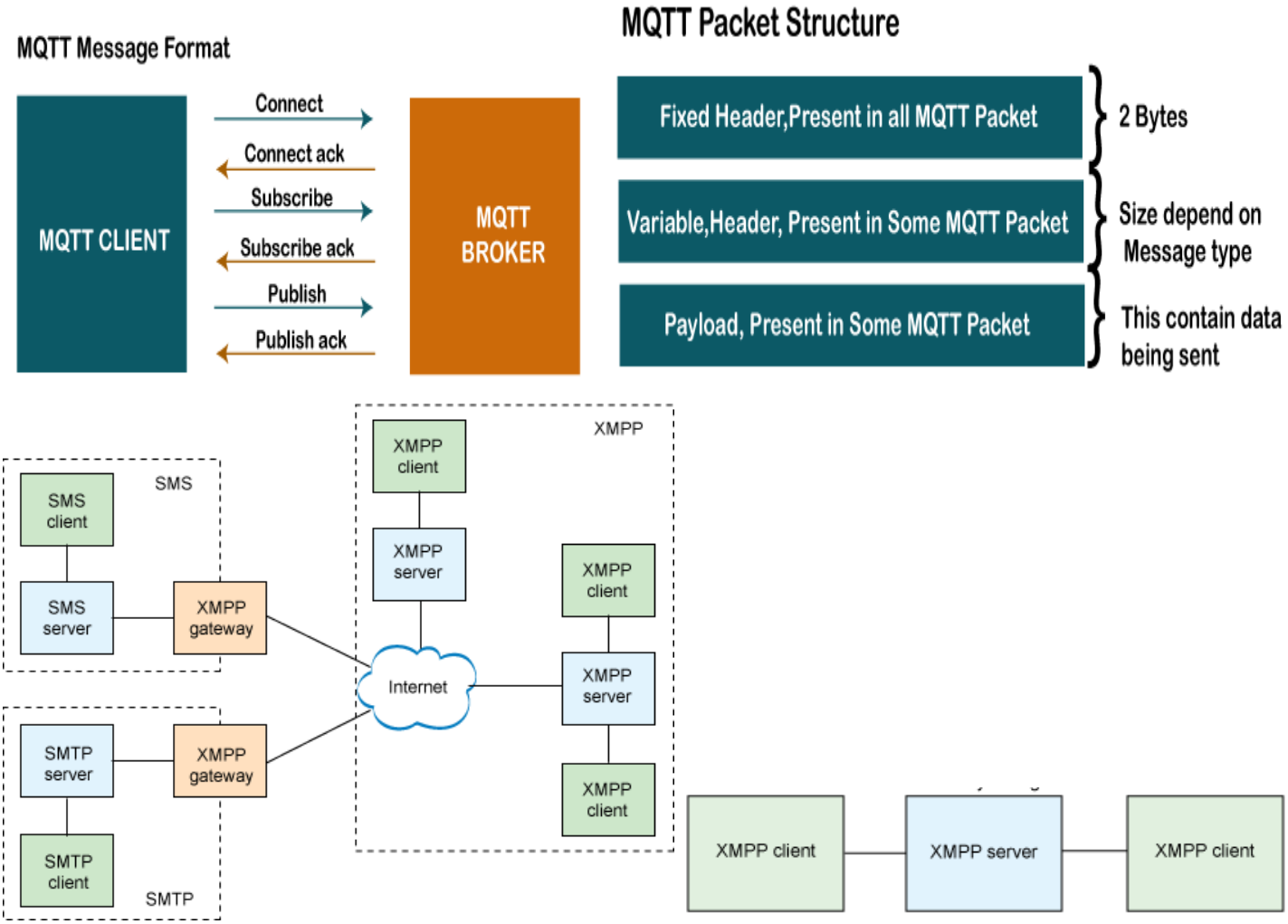

Internet of Things Reference Model
Levels
7 Collaboration & Processes
(Involving People & Business Processes)
6 Application
(Reporting, Analytics, Control)
5 Data Abstraction
(Aggregation & Access)
4 Data Accumulation
(Storage)
3 Edge Computing
(Data Element Analysis & Transformation)
2 Connectivity
(Communication & Processing Units)
1 Physical Devices & Controllers
(The "Things" in IoT)
Center
Edge
Sensors, Devices, Machines,
Intelligent Edge Nodes of all types
The model is based on "Integrated Security & Management"
Data at Rest
IT
OT
Data in Motion
The model is based on "Information Flow"



Layer	Name	Protocols
Layer 7	Application	SMTP, HTTP, FTP, POP3, SNMP
Layer 6	Presentation	MPEG, ASCH, SSL, TLS
Layer 5	Session	NetBIOS, SAP
Layer 4	Transport	TCP, UDP
Layer 3	Network	IPV5, IPV6, ICMP, IPSEC, ARP, MPLS.
Layer 2	Data Link	RAPA, PPP, Frame Relay, ATM, Fiber Cable, etc.
Layer 1	Physical	RS232, 100BaseTX, ISDN, 11.

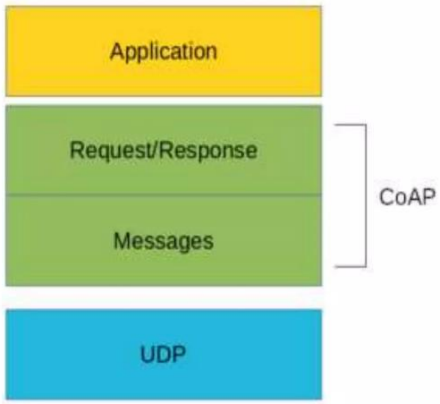




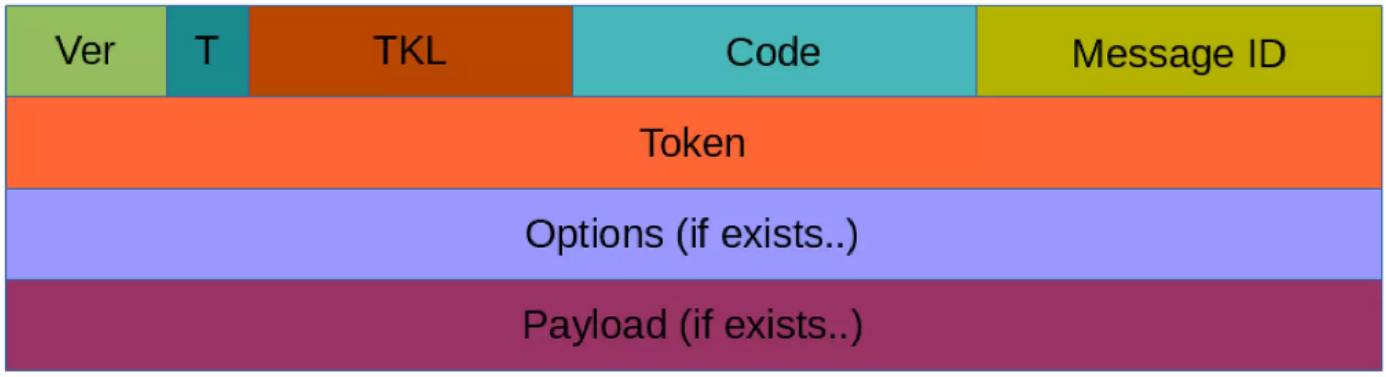


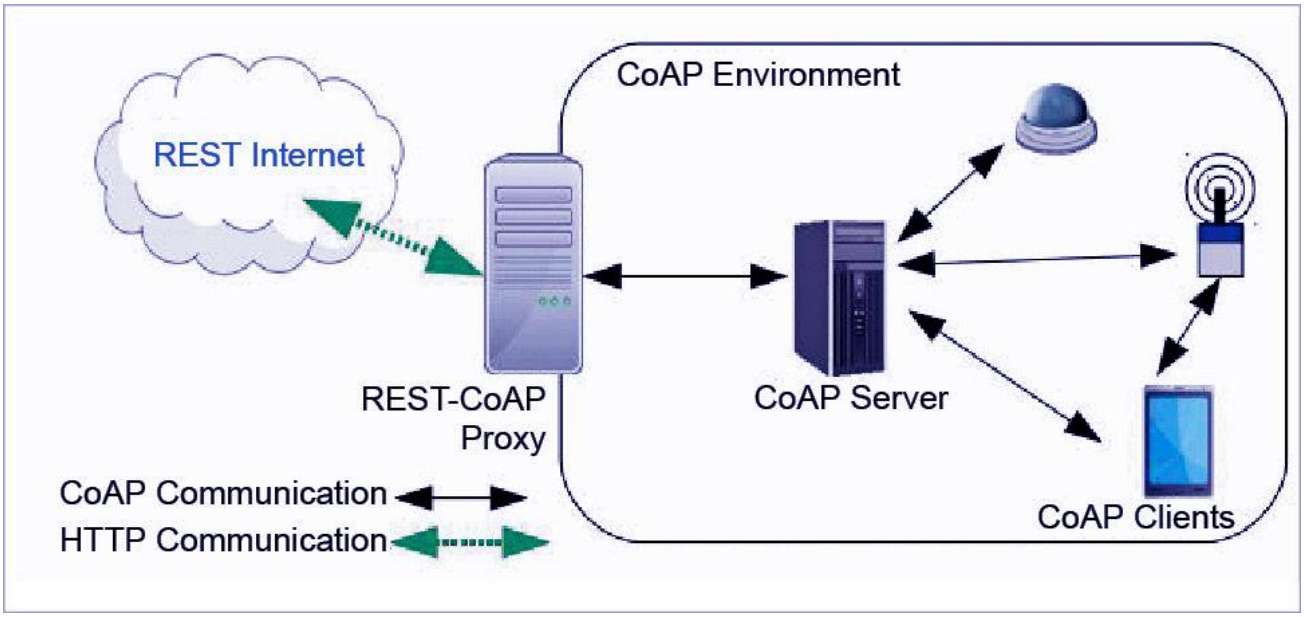
CoAP Structure

- There are two different layers that make CoAP protocol:
  1)Messages
  2)Request/Response.
- The **Messages layer** deals with UDP and with asynchronous messages.
- The **Request/Response layer** manages request/response interaction based on request/response messages.

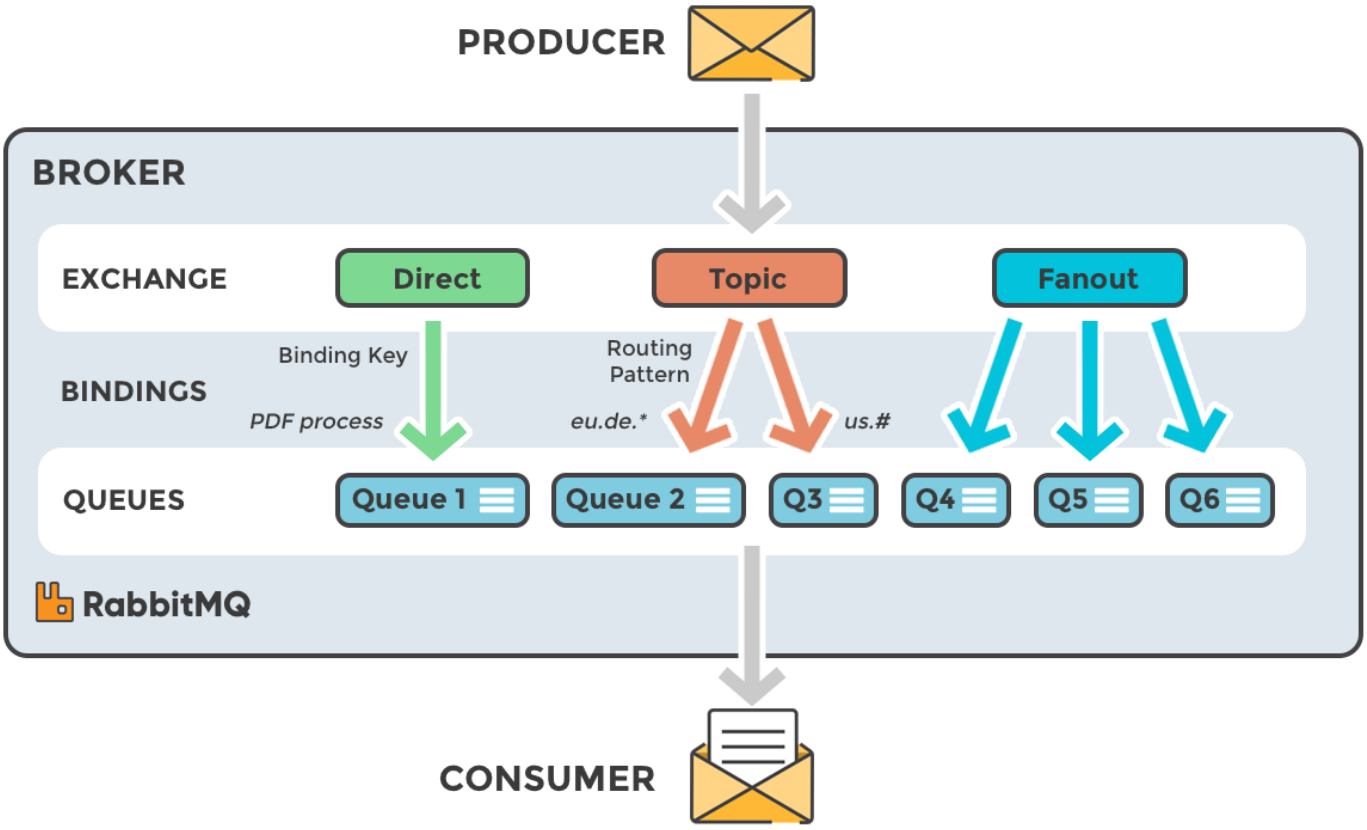


CoAP Message Format

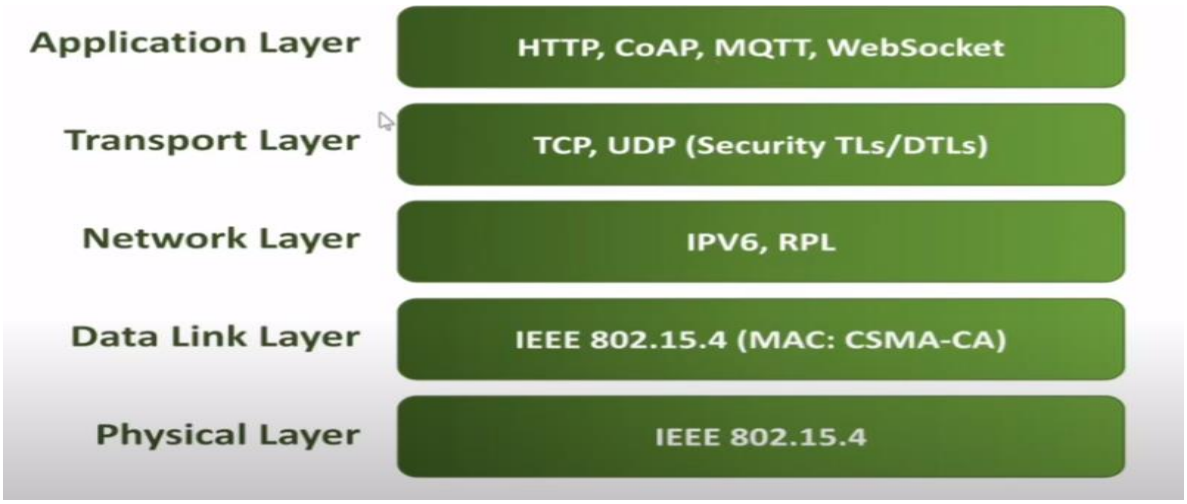


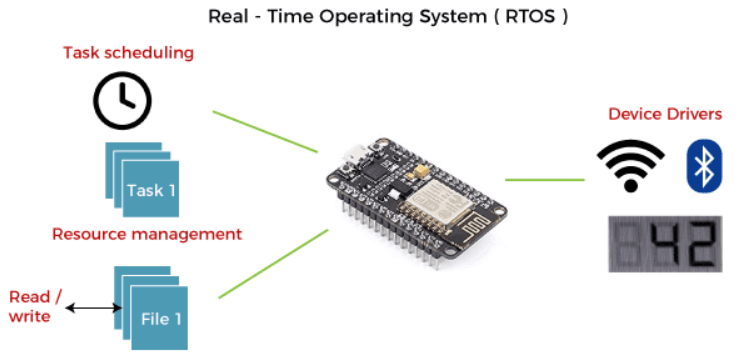


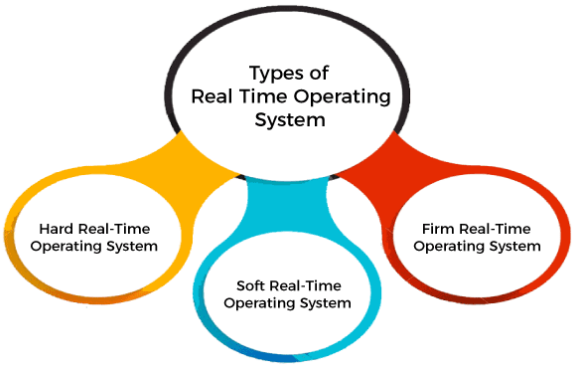
AMQP Architecture


PRODUCER
BROKER
EXCHANGE
Direct
Topic
Fanout
BINDINGS
Binding Key
PDF process
Routing Pattern
eu.de.*
us.#
QUEUES
Queue 1
Queue 2
Q3
Q4
Q5
Q6
RabbitMQ
CONSUMER


Architecture of 6LOWPAN
IoT devices with
Wireless Communication
Edge Router
Wired/Wireless Communication
Server, Cloud Storage, Server Applications, Internet etc.


Application Layer
HTTP, CoAP, MQTT, WebSocket
Transport Layer
TCP, UDP (Security TLs/DTLs)
Network Layer
IPV6, RPL
Data Link Layer
IEEE 802.15.4 (MAC: CSMA-CA)
Physical Layer
IEEE 802.15.4





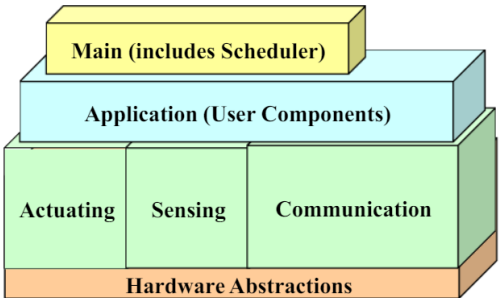
TinyOS Overview

Application = scheduler + graph of components
- Compiled into one executable

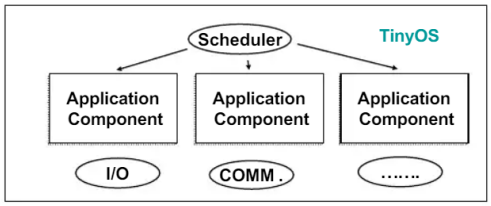
Event-driven architecture

Single shared stack

No kernel/user space differentiation



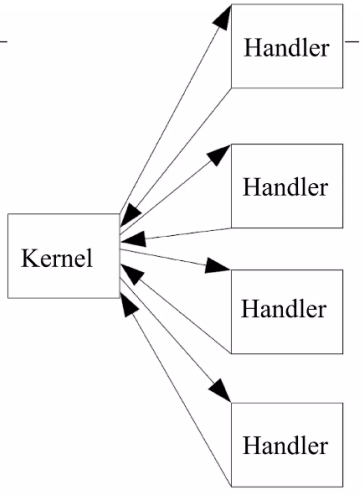
TinyOS Architecture Overview (1)

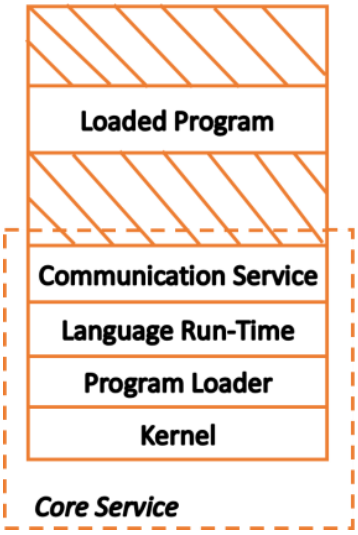


NO Kernel ➡ Direct hardware manipulation
NO Process management ➡ Only one process on the fly.
NO Virtual memory ➡ Single linear physical address space
NO Dynamic memory allocation ➡ Assigned at compile time
NO Software signal or exception ➡ Function Call instead

Event-driven

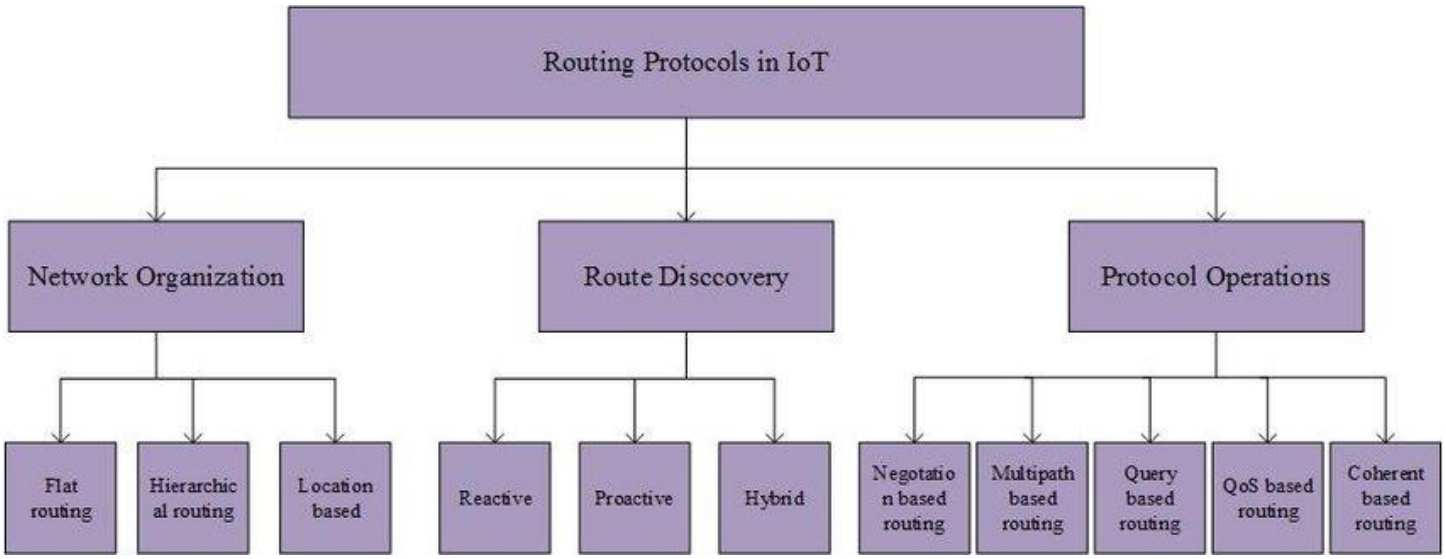
- Event-driven (TinyOS)
  - Processes do not run without events
  - Event occurs: kernel invokes event handler
  - Event handler runs to completion (explicit `return;`)



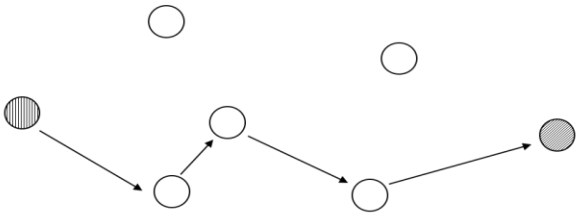


Loaded Programs Can be easily updated

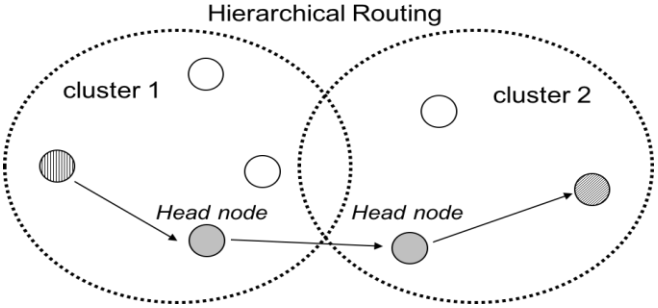
Core : Single binary Usually never modified

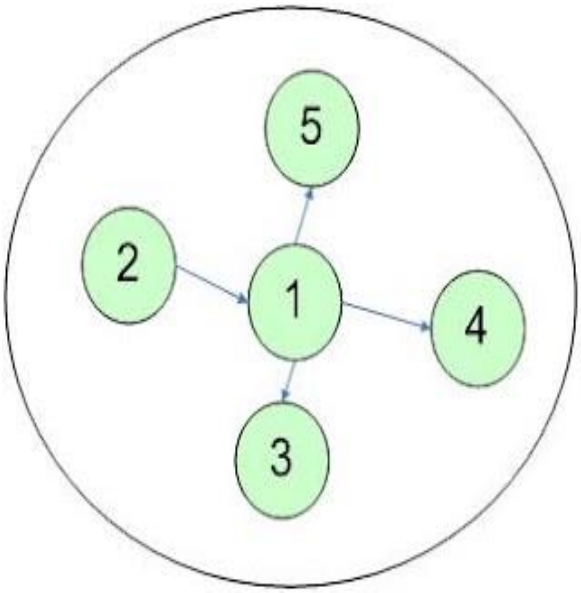


Flat Routing

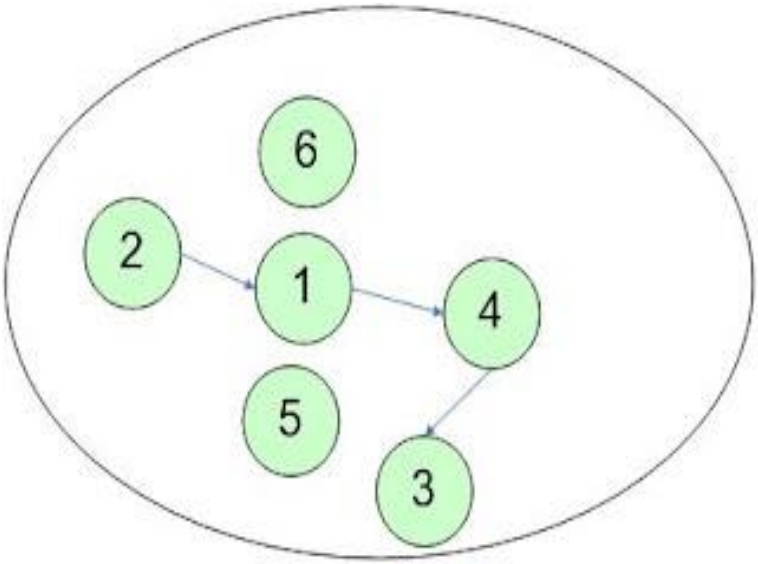


Hierarchical Routing



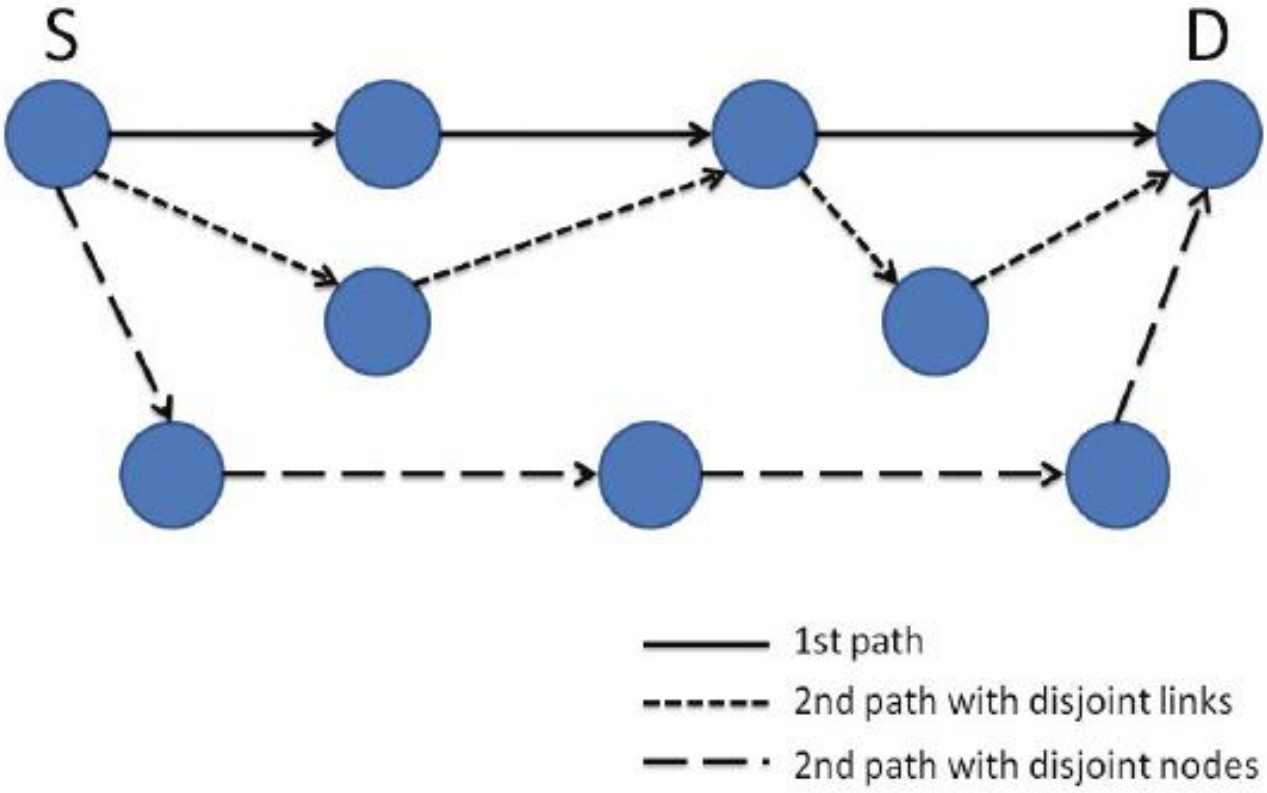


Flooding



Gossiping

Multipath routing



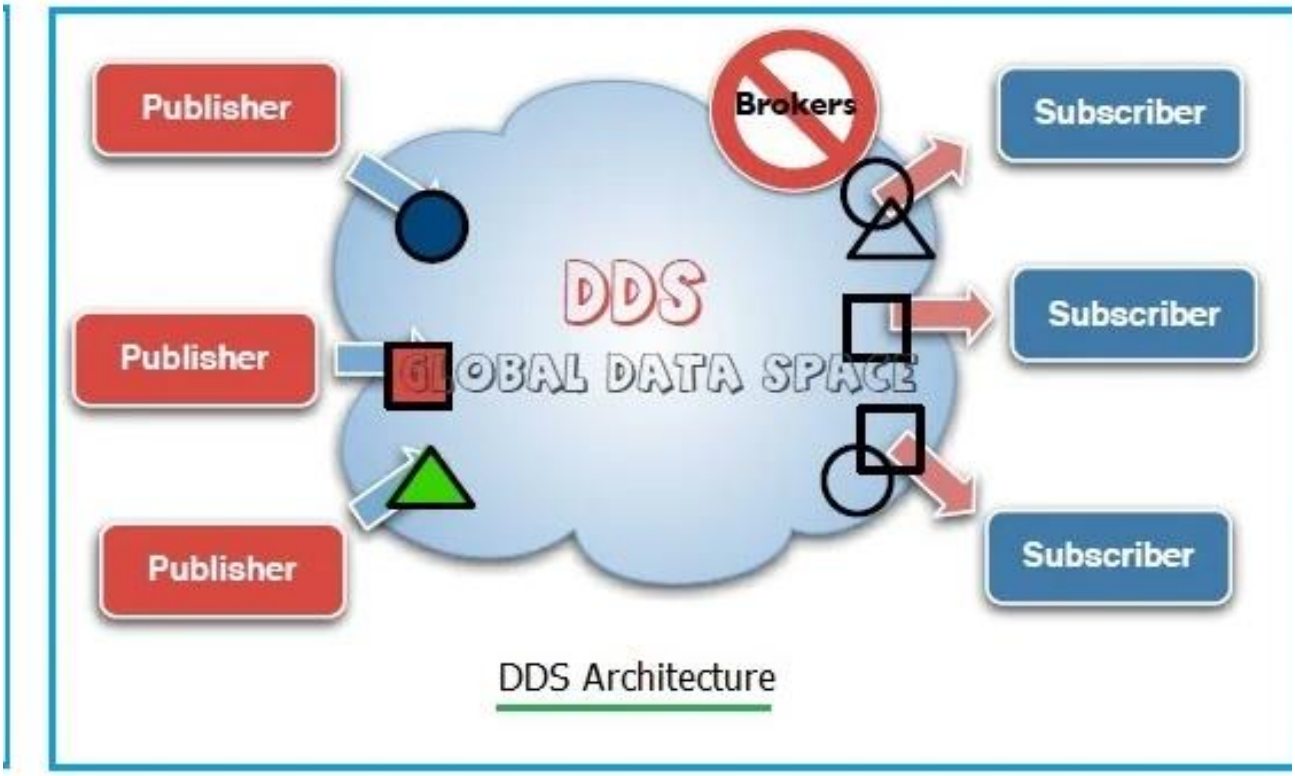
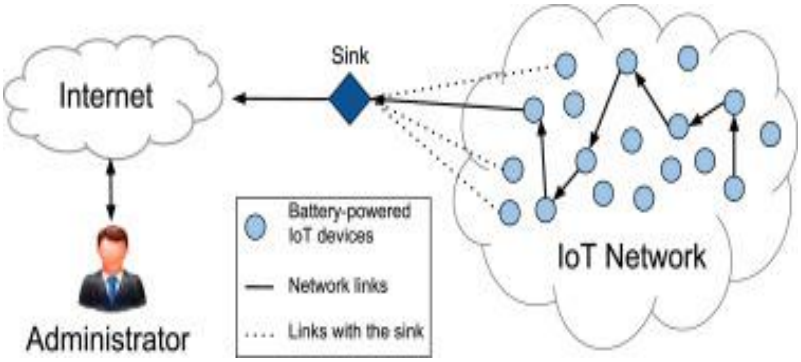
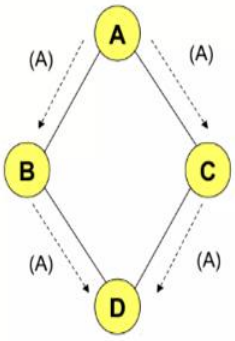


Figure-2: DDS Protocol Architecture

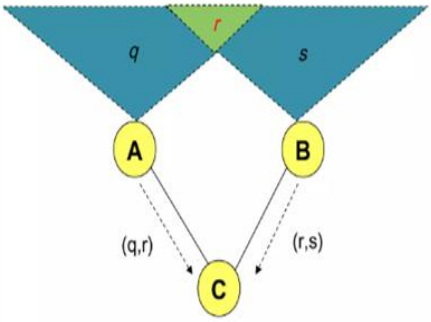


Implosion



The implosion problem. In this graph, node A starts by flooding its data to all of its neighbors. Two copies of the data eventually arrive at node D. The system energy waste energy and bandwidth in one unnecessary send and receive.

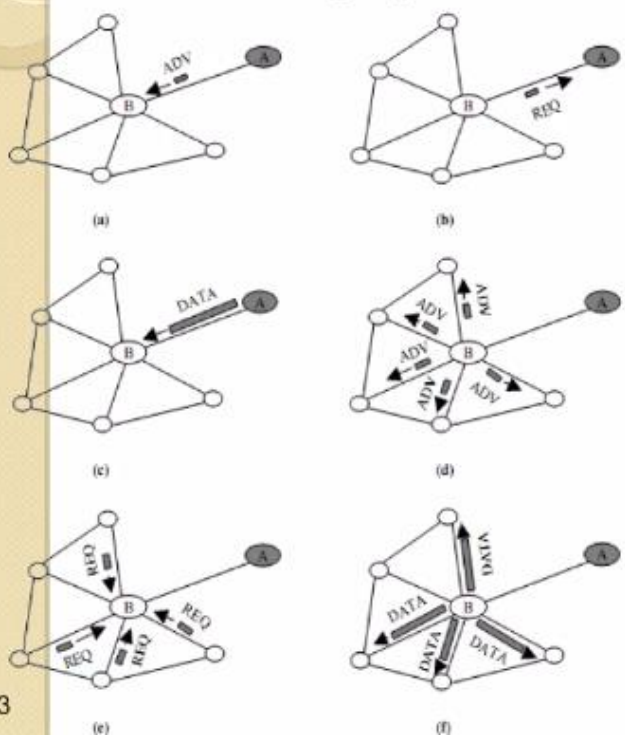
Overlap



The overlap problem. Two sensors cover an overlapping geographic region. When these sensors flood their data to node C, C receives two copies of the data marked r.

SPIN-1 : 3-Stage Handshake Protocol (SPIN-PP)

- Simple handshake protocol for disseminating data through a lossless network
- Work in three stages (ADV-REQ-DATA)



Node A starts by advertising its data to node B (a).

Node B responds by sending a request to node A (b).

After receiving the requested data (c), node B then sends out advertisement to its neighbors (d), who in turn send requests back to B (e, f).

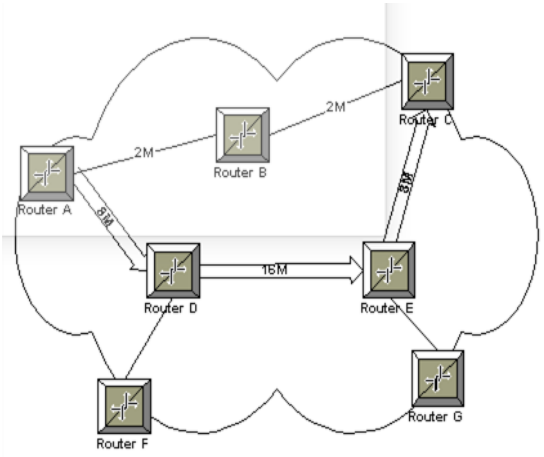
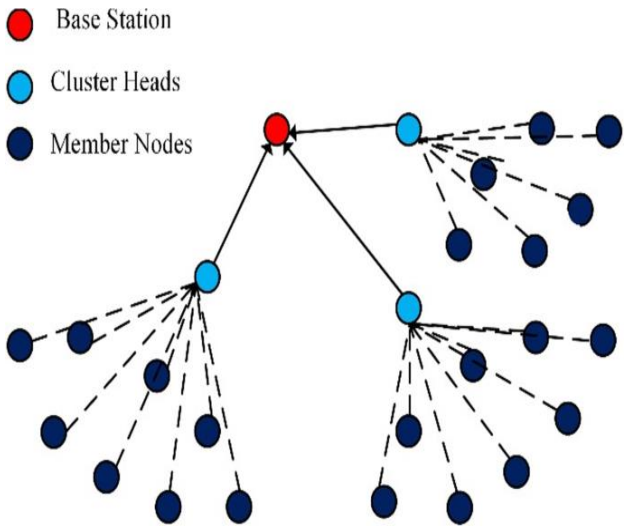
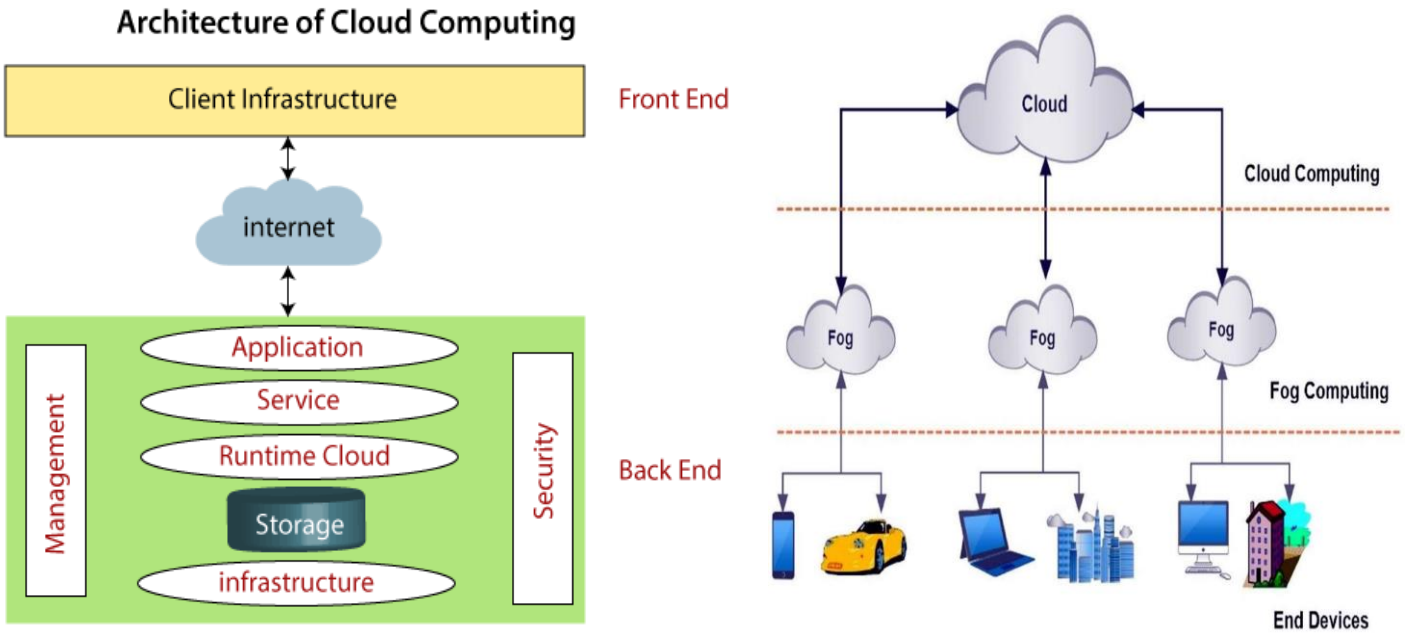


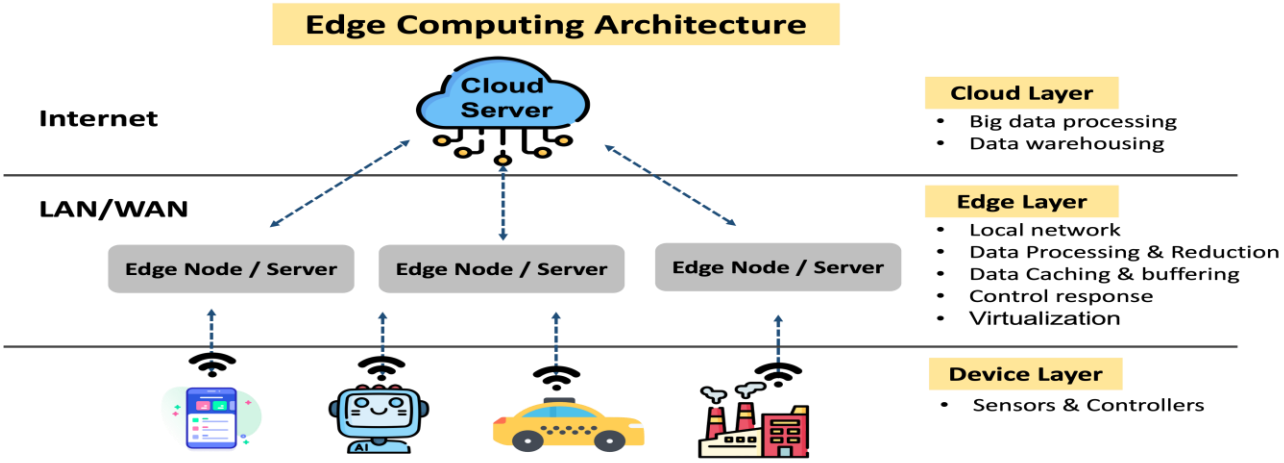
Figure 1: QoS-based routing example



Leach

Module 6





Figure：Edge computing architecture overview
Source：The research team